

TIBCO Business Studio™

Process Modeling

*Software Release 3.5
April 2011*

Important Information

SOME TIBCO SOFTWARE EMBEDS OR BUNDLES OTHER TIBCO SOFTWARE. USE OF SUCH EMBEDDED OR BUNDLED TIBCO SOFTWARE IS SOLELY TO ENABLE THE FUNCTIONALITY (OR PROVIDE LIMITED ADD-ON FUNCTIONALITY) OF THE LICENSED TIBCO SOFTWARE. THE EMBEDDED OR BUNDLED SOFTWARE IS NOT LICENSED TO BE USED OR ACCESSED BY ANY OTHER TIBCO SOFTWARE OR FOR ANY OTHER PURPOSE.

USE OF TIBCO SOFTWARE AND THIS DOCUMENT IS SUBJECT TO THE TERMS AND CONDITIONS OF A LICENSE AGREEMENT FOUND IN EITHER A SEPARATELY EXECUTED SOFTWARE LICENSE AGREEMENT, OR, IF THERE IS NO SUCH SEPARATE AGREEMENT, THE CLICKWRAP END USER LICENSE AGREEMENT WHICH IS DISPLAYED DURING DOWNLOAD OR INSTALLATION OF THE SOFTWARE (AND WHICH IS DUPLICATED IN THE LICENSE FILE) OR IF THERE IS NO SUCH SOFTWARE LICENSE AGREEMENT OR CLICKWRAP END USER LICENSE AGREEMENT, THE LICENSE(S) LOCATED IN THE "LICENSE" FILE(S) OF THE SOFTWARE. USE OF THIS DOCUMENT IS SUBJECT TO THOSE TERMS AND CONDITIONS, AND YOUR USE HEREOF SHALL CONSTITUTE ACCEPTANCE OF AND AN AGREEMENT TO BE BOUND BY THE SAME.

This document contains confidential information that is subject to U.S. and international copyright laws and treaties. No part of this document may be reproduced in any form without the written authorization of TIBCO Software Inc.

TIBCO, The Power of Now, TIBCO ActiveMatrix, TIBCO Business Studio, TIBCO iProcess, TIBCO BusinessWorks, TIBCO PageBus are either registered trademarks or trademarks of TIBCO Software Inc. in the United States and/or other countries.

EJB, Java EE, J2EE, JMS and all Java-based trademarks and logos are trademarks or registered trademarks of Sun Microsystems, Inc. in the U.S. and other countries.

The BPMN logo is a trademark or registered trademark of the Object Management Group, Inc. in the United States and other countries.

XPDL and Powered by XPDL are trademarks of the Workflow Management Coalition.

All other product and company names and marks mentioned in this document are the property of their respective owners and are mentioned for identification purposes only.

THIS SOFTWARE MAY BE AVAILABLE ON MULTIPLE OPERATING SYSTEMS. HOWEVER, NOT ALL OPERATING SYSTEM PLATFORMS FOR A SPECIFIC SOFTWARE VERSION ARE RELEASED AT THE SAME TIME. SEE THE README.TXT FILE FOR THE AVAILABILITY OF THIS SOFTWARE VERSION ON A SPECIFIC OPERATING SYSTEM PLATFORM.

THIS DOCUMENT IS PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NON-INFRINGEMENT.

THIS DOCUMENT COULD INCLUDE TECHNICAL INACCURACIES OR TYPOGRAPHICAL ERRORS. CHANGES ARE PERIODICALLY ADDED TO THE INFORMATION HEREIN; THESE CHANGES WILL BE INCORPORATED IN NEW EDITIONS OF THIS DOCUMENT. TIBCO SOFTWARE INC. MAY MAKE IMPROVEMENTS AND/OR CHANGES IN THE PRODUCT(S) AND/OR THE PROGRAM(S) DESCRIBED IN THIS DOCUMENT AT ANY TIME.

THE CONTENTS OF THIS DOCUMENT MAY BE MODIFIED AND/OR QUALIFIED, DIRECTLY OR INDIRECTLY, BY OTHER DOCUMENTATION WHICH ACCOMPANIES THIS SOFTWARE, INCLUDING BUT NOT LIMITED TO ANY RELEASE NOTES AND "READ ME" FILES.

Copyright © 2004-2011 TIBCO Software Inc. ALL RIGHTS RESERVED.

TIBCO Software Inc. Confidential Information

Contents

Tables	xi
Preface	xiii
Changes from the Previous Release of this Guide	xiv
Related Documentation	xvi
Third-Party Documentation	xvi
Typographical Conventions	xvii
Connecting with TIBCO Resources	xix
How to Join TIBCOCommunity	xix
How to Access All TIBCO Documentation	xix
How to Contact TIBCO Support	xix
Chapter 1 Overview	1
Who Should Use TIBCO Business Studio?	2
TIBCO Business Studio Modes	2
Process Modeling Methodology	4
Process Components	5
Example	6
Pageflow Process Modeling	7
About Projects, Packages and Processes	9
Projects	9
Packages	10
Processes	10
Asset Types	11
Standards Support	12
Business Process Modeling Notation (BPMN)	12
XML Process Definition Language (XPDL)	12
Capabilities and Perspectives	13
Chapter 2 Getting Started for Analysts	15
Modelled Application Archive	16
Creating and Accessing MAAs	17
The TIBCO Business Studio Ribbon	18

Chapter 3 Getting Started for Designers	21
Creating a Project, Package and Process	22
Working with Processes	27
Adding Objects	27
Connecting Objects	30
Selecting Objects	32
Finding Objects	32
Customizing Palette Favorites	33
Chapter 4 Working with Projects, Packages, and Processes	35
Migrating Processes Created in Previous Versions	36
Migrating from Version 1.1	36
Migrating from Version 2.x or 3.x	37
Migrating Business Assets	37
Creating a Package	40
Copying a Package	41
Creating a Business Process	42
Creating a Pageflow Process	46
Deleting a Process	49
Creating Extended Attributes	51
Creating Project References	52
Creating References in the Project Properties	52
Creating Project References in a Selection Dialog	53
Working with Assets	54
Business Process Assets	54
Forms Assets	54
Organization Model Assets	54
Service Assets	55
Business Object Model Assets	55
Business Assets	56
Working With Special Folders	58
Enabling and Disabling Special Folders	59
Managing Project Lifecycle	60
Changing Project Lifecycle Settings	60
Refactoring Lifecycle Settings	61
Importing and Exporting Projects	63
Importing Existing Projects into Workspace	63
Importing Existing MAA Files into Workspace	65
Importing Archive Files into Workspace	66
Exporting Projects to Archive File	66
Exporting Projects to File System	68

Exporting Projects to Modelled Application Archive (MAA)	68
Using Process Fragments	69
Migrating Fragments from Previous Versions	69
Using the Predefined Fragments	69
Creating Custom Fragments	70
Using Custom Fragments	71
Importing Custom Fragments	71
Using Source Control	72
Using Subversion with TIBCO Business Studio	72
Creating a Project from an Existing Project in Source Control	73
Browsing SVN Repositories	73
Deleting Projects from SVN Repositories with Asset Central	74
Comparing Different Versions of a Process Package	75
Emailing Processes	80
Creating Documentation	81
To Create Documentation from within TIBCO Business Studio	81
To Create Documentation from the Command Line	82
To Generate Documentation On Demand	82
To View Documentation	83
Using Presentation Channels	84
Editing Email Attributes at Workspace Level	84
Editing at Project Level	86
Creating an Alternative Email Template	87
Chapter 5 Working with Activities	91
Activity and Task Overview	92
User Tasks	92
Manual Tasks	93
Service Tasks	93
Script Tasks	93
Send Tasks	94
Receive Tasks	94
Reference Tasks	95
Request Response Operation with Send and Receive Tasks	97
Creating Scripts	98
Sub-Processes	100
Embedded	100
Reusable	100
Transactions	101
Activity Markers	102
Working with User Tasks and Pageflow Processes	103

Selecting a Pageflow Process	103
Creating a Pageflow Process From a User Task	104
Synchronizing Parameters With a Pageflow Process	106
Mandatory Parameters and Pageflow Processes	107
Resource Patterns and Work Distribution	108
Resources	108
Distribution Strategy	109
Piling	109
Separation of Duties	109
Retain Familiar	112
Chained Execution	116
Making Processes Easier to Follow	118
Associations	118
Data Objects	118
Text Annotations	119
Groups	120
Swimlanes	121
Lanes	121
Pools	123
Chapter 6 Controlling Flow	125
Message Flows	126
Sequence Flows	127
Using Loops	128
Creating a Standard Loop	128
Creating a Multi-Instance Loop	129
Gateways	134
Order of Flow Evaluation	134
Exclusive (XOR)	136
Inclusive (OR)	136
Complex	137
Parallel (AND)	138
Chapter 7 Working with Events	139
About Events	140
Throw and Catch Events	140
Triggers and Results	141
Summary	142
Start Events	143
Intermediate Events	144
End Events	146

Configuring Timer Event Scripts	147
Using Throw and Catch Signal Events	149
Using Throw and Catch Error Events	151
Configuring Error Events	152
Fixing Invalid Error Events	155
Mapping Error Data	155
Configuring Compensation Events	157
Implementing Message Events	158
Receiving a One-Way Request Operation	159
Sending a One-Way Request Operation	160
Receiving a Request-Response Operation	160
Chapter 8 Working with Process Data	163
Creating Data Fields and Parameters	164
Adding Data Fields or Parameters to a Package or Process	165
Adding Data Fields to an Activity	168
Handling Incoming Messages Using Correlation Data	169
Creating Correlation Data	170
Creating Declared Types	172
Associating Process Data with Events and Tasks	173
Associating Correlation Data with an Event or Task	176
Setting Event and Task Visibility (Private and Public)	178
Using Process Data to Create Tasks	179
Creating Participants	180
Basic Types	180
Participants from the Organization Model	180
Creating a Participant	181
Associating Participants with Activities	182
Exporting Participants to an Organization Model	185
Participant to Organization Model Mapping	186
Using Participants to Create Tasks	187
Deleting Data Fields, Parameters, Participants, and Type Declarations	188
Chapter 9 Using Sub-Processes and Process Interfaces	189
Reusable Sub-Processes	190
Expanding a Sub-Process	193
Embedded Sub-Processes	194
Local Data Fields	194
Refactoring Sub-Processes	197
Creating a New Embedded Sub-Process	197

Creating a New Reusable Sub-Process	199
Inline Sub-Process Content	204
Inline Sub-Processes (Package Optimization)	208
Creating an Inline Sub-Process	209
Inline Sub-Process in Detail	211
Process Objects	211
Data Fields and Mapped Parameters	211
Sub-Process Data Fields and Unmapped Formal Parameters	212
Sub-Process Participants And Type Declarations	214
Criteria for Inline Sub-Processes	215
Problems that Prevent Optimization	215
Optimization Categories	216
Process Interfaces	218
Creating a Process Interface	221
Parts of a Process Interface	221
Error Codes and Process Interfaces	225
Making a Sub-Process Call to a Process Interface	228
Modifying a Process Interface	231
Using a Dynamic Sub-Process	232
Creating a Dynamic Sub-Process	232
Automatically Passing Different Data To and From Sub-Process Instances	233
Chapter 10 Integrating with Other Applications	235
Destinations and Validation	236
Correcting Validation Errors	236
Setting the Validation Preferences	237
Configuring the Problems View	238
Integrating with ARIS Software	240
Creating Custom Symbols in ARIS (Optional)	240
Using Custom Symbols in TIBCO Business Studio	240
Exporting from ARIS	241
Importing ARIS XML Files into TIBCO Business Studio	242
Object Mappings — ARIS Software	244
Extended Attributes	246
Integrating with Other Software	247
Chapter 11 Reference	249
Project Explorer View	250
Problems View	251
Outline View	252

Fragments View	253
Properties View	254
Labels and Names	254
Process Properties	255
Activity Properties	257
Gateway Properties	262
Sequence Flow Properties	263
Event Properties	264
Lane Properties	265
Cheat Sheets View	266
Process Editor	267
Palette	267
Alignment Tools	269
Chapter 12 Tips and Tricks	271
Process Editor	272
Workbench	274
Problems View	275
Index	277

Tables

Table 1	General Typographical Conventions	xvii
Table 2	Summary of Event Types	142
Table 3:	ARIS and TIBCO Business Studio Object Mappings	244
Table 4:	Process Property Reference	255
Table 5:	Activity Property Reference	257
Table 6:	Gateway Property Reference	262
Table 7:	Sequence Flow Property Reference	263
Table 8:	Event Property Reference	264
Table 9:	Lane Property Reference	265
Table 10:	Palette Tool Reference	267

Preface

This guide is intended for business analysts and describes using the process modeling capabilities of TIBCO Business Studio. No previous software experience is necessary, but a familiarity with Business Process Modeling Notation (BPMN) is useful (see <http://www.bpmn.org>).



- This guide does not cover implementation, export, or deployment. For more information about how to develop processes for a specific runtime environment, see the appropriate implementation guide.
- The features described in this guide are available in the Business Analysis capability of the Modeling perspective (for more information about perspectives, see [Capabilities and Perspectives on page 13](#)).

Topics

- [Changes from the Previous Release of this Guide, page xiv](#)
- [Related Documentation, page xvi](#)
- [Typographical Conventions, page xvii](#)
- [Connecting with TIBCO Resources, page xix](#)

Changes from the Previous Release of this Guide

This section itemizes the major changes from the previous release of this guide.

Studio for Analysts

A new chapter [Getting Started for Analysts, page 15](#) describes an addition to the installation of TIBCO Business Studio. It provides a cut-down version of the product for analysts to use.

Dynamic sub-processes

Dynamic sub-processes are supported (as in TIBCO iProcess Engine) using a sub-process identifier of a scalar or array. See [Using a Dynamic Sub-Process on page 232](#).

Adding Additional Instances to a Process when it is Running

When a Process is running, you may want to add additional instances (for example, to meet an unplanned request from a customer for additional goods). See [Adding Additional Instances to a Multi-Instance Loop Task \(whilst that task is in progress\) \(WCP-15\) on page 131](#).

Creating an Alternative Email Template

You can create your own email template which can contain organization-specific information, with your own logos and corporate style. It will be used in the email sent to the user after deployment when you have pushed an email task to the Workspace. See [Creating an Alternative Email Template on page 87](#).

Process Package Viewer

You can compare the local Process Package you are working on with another version to identify any changes you have made since the last time you saved that Process Package or any changes made by another user in the Subversion copy since you last checked it out. See [Comparing Different Versions of a Process Package on page 75](#).

Email Error Events

You can catch email errors by using a Catch Error Event attached to an email service task. See [Configuring Error Events on page 152](#).

Importing and Exporting to Modelled Application Archive (MAA) Files

MAA files can now be imported, and projects can be exported to MAA files. See [Importing Existing MAA Files into Workspace on page 65](#) and [Exporting Projects to Modelled Application Archive \(MAA\) on page 68](#).

Related Documentation

This section lists documentation resources you may find useful.

Third-Party Documentation

The Eclipse help contains useful information on the Workbench and the Eclipse user interface.

TIBCO Business Studio supports the following standards:

- Business Process Modeling Notation (BPMN)
<http://www.bpmn.org>
- XML Process Definition Language (XPDL)
<http://www.wfmc.org/standards/xpdl.htm>

Typographical Conventions

The following typographical conventions are used in this manual.

Table 1 General Typographical Conventions

Convention	Use
<i>TIBCO_HOME</i> <i>ENV_HOME</i> <i>STUDIO_HOME</i>	Many TIBCO products must be installed within the same home directory. This directory is referenced in documentation as <i>TIBCO_HOME</i>. The default value of <i>TIBCO_HOME</i> depends on the operating system. For example, on Windows systems, the default value is C:\tibco. Other TIBCO products are installed into an <i>installation environment</i>. Incompatible products and multiple instances of the same product are installed into different installation environments. An environment home directory is referenced in documentation as <i>ENV_HOME</i>. The default value of <i>ENV_HOME</i> depends on the operating system. For example, on Windows systems the default value is C:\tibco. TIBCO Business Studio installs into a directory within <<i>TIBCO_HOME</i>>. This directory is referenced in documentation as <i>STUDIO_HOME</i>. The default value of <i>STUDIO_HOME</i> depends on the operating system. For example on Windows systems, the default value is C:\Program Files\TIBCO\studio-bpm-35.
code font	Code font identifies commands, code examples, filenames, pathnames, and output displayed in a command window. For example: Use MyCommand to start the foo process.
bold code font	Bold code font is used in the following ways: • In procedures, to indicate what a user types. For example: Type admin. • In large code samples, to indicate the parts of the sample that are of particular interest. • In command syntax, to indicate the default parameter for a command. For example, if no parameter is specified, MyCommand is enabled: MyCommand [enable disable]

Table 1 General Typographical Conventions (Cont'd)

Convention	Use
<i>italic font</i>	Italic font is used in the following ways: • To indicate a document title. For example: See <i>TIBCO ActiveMatrix BusinessWorks Concepts</i>. • To introduce new terms. For example: A portal page may contain several portlets. <i>Portlets</i> are mini-applications that run in a portal. • To indicate a variable in a command or code syntax that you must replace. For example: <code>MyCommand</code> <i>PathName</i>
Key combinations	Key name separated by a plus sign indicate keys pressed simultaneously. For example: Ctrl+C. Key names separated by a comma and space indicate keys pressed one after the other. For example: Esc, Ctrl+Q.
	The note icon indicates information that is of special interest or importance, for example, an additional action required only in certain circumstances.
	The tip icon indicates an idea that could be useful, for example, a way to apply the information provided in the current section to achieve a specific result.
	The warning icon indicates the potential for a damaging situation, for example, data loss or corruption if certain steps are taken or not taken.

Connecting with TIBCO Resources

How to Join TIBCOCommunity

TIBCOCommunity is an online destination for TIBCO customers, partners, and resident experts, a place to share and access the collective experience of the TIBCO community. TIBCOCommunity offers forums, blogs, and access to a variety of resources. To register, go to <http://www.tibcommunity.com>.

How to Access All TIBCO Documentation

After you join TIBCOCommunity, you can access the documentation for all supported product versions here:

<http://docs.tibco.com/TibcoDoc>

How to Contact TIBCO Support

For comments or problems with this manual or the software it addresses, please contact TIBCO Support as follows.

- For an overview of TIBCO Support, and information about getting started with TIBCO Support, visit this site:

<http://www.tibco.com/services/support>

- If you already have a valid maintenance or support contract, visit this site:

<https://support.tibco.com>

Entry to this site requires a user name and password. If you do not have a user name, you can request one.

Chapter 1 **Overview**

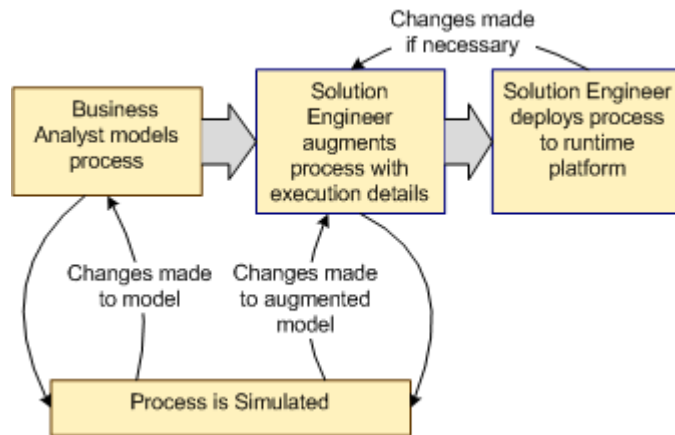
This chapter provides an overview of how TIBCO Business Studio is used to model processes.

Topics

- [Who Should Use TIBCO Business Studio?, page 2](#)
- [Process Modeling Methodology, page 4](#)
- [Pageflow Process Modeling, page 7](#)
- [About Projects, Packages and Processes, page 9](#)
- [Standards Support, page 12](#)
- [Capabilities and Perspectives, page 13](#)

Who Should Use TIBCO Business Studio?

TIBCO Business Studio is for business analysts and solution designers, including those responsible for the analysis, design, simulation, implementation, and deployment of business processes. The following diagram shows how TIBCO Business Studio is intended to be used:



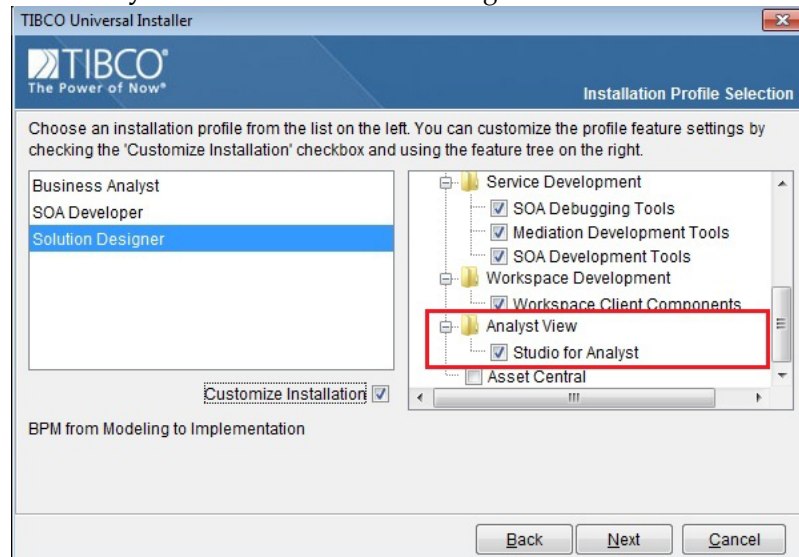
TIBCO Business Studio Modes

TIBCO Business Studio can be launched in two modes:

- Business Analyst** enables the business analyst to model the business processes and access all the artifacts from a single file. This mode is suited to users doing high level or abstract modeling. The ribbon-based toolbar provides a familiar, easy to use interface that supports this type of modeling. To launch TIBCO Business Studio in Business Analyst mode, go to **Start > Programs > TIBCO > environment_name > TIBCO Business Studio 3.5 BPM Edition > Studio for Analysts**.
- Solution Designer** provides a project based view and enables the solution designer to add details to and manage the processes at a project level. This mode is suited to users who are more comfortable with developer tools, and who need access to all files related to the project. To launch TIBCO Business Studio in Solution Designer mode, go to **Start > Programs > TIBCO > environment_name > TIBCO Business Studio 3.5 BPM Edition > Studio for Designers**.



Studio for Analysts is installed by default. To unselect the Studio for Analysts feature during installation, choose the **Customize Installation** checkbox on the **Installation Profile Selection** dialog and uncheck the feature **Analyst View > Studio for Analyst** in the feature tree on the right.



Process Modeling Methodology

Modeling a process can be achieved in several different ways, however the following general approach reflects best practices:

Task A Define the "As is" Process

1. Interview business end users about their current practices.
2. Capture the process flow (either on paper first or directly in modeling tool).
3. Capture the process relevant data (either on paper first or directly in modeling tool).



Set the appropriate *destination environment* on the process (this specifies the runtime environment where the process will be executed). This can be done either when first creating the process, or at any time prior to implementation. Setting the destination environment when the process is first created avoids error messages and warnings associated with modeling constructs that cannot be executed in the runtime environment.

4. Attach the process-relevant data at key points (for example, at decision points or certain activities).
5. Simulate and analyze to ensure that the "As is" process is an accurate representation of the current process.

Task B Define the "To be" Process (Optional)

This is an iterative exercise in which you:

1. Propose optimizations (process changes and new automation of existing processes).
2. Simulate to validate changes or to quantify estimated savings.

Task C Define the Business Object Model and Organization Model (Optional)

Define a *business object model* that defines key business terms specific to your corporate environment (for example, in an insurance environment, a claim, claimant, and so on). This can be used as an analysis tool.

Task D Hand Over for Implementation (Optional)



If you have been using Studio for Analysts to create the process, you must switch to Studio for Designers to complete the implementation.

If the process is to be executed in a BPM environment, you should do the following:

1. If it has not already been set, set the appropriate *destination environment* (this specifies the target environment where the process will be executed).
2. Check the Problems view for any warnings or errors in the process.
3. Hand the process off to the solution engineer for implementation. The solution engineer will underpin the process with the necessary details (such as calls to web services and so on) that will enable the process to execute in the specified destination environment.

Task E Deployment

Deployment is part of the software development cycle (design, deploy, execute). After preparing the software, some transformation, packaging, physical delivery, configuration and initialization takes place. All of these, some of which may be optional, are aspects of deployment.

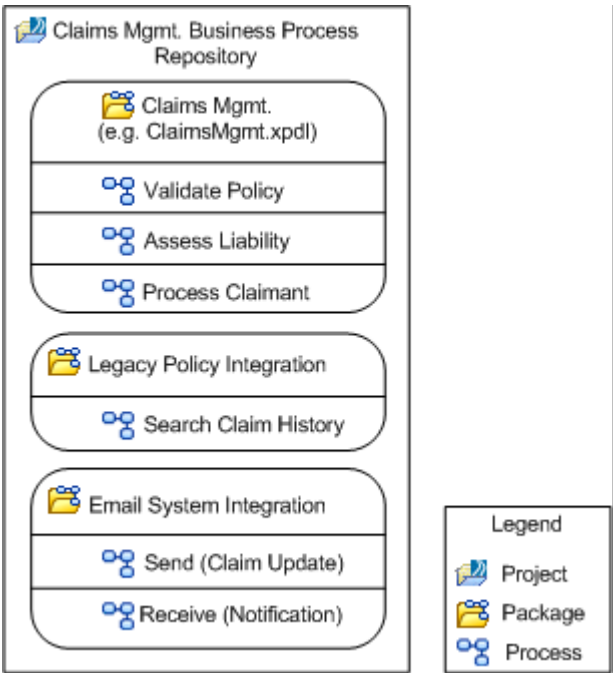
For more information about deployment, see the destination-specific implementation guide.

Process Components

Process components represent reusable building blocks that encapsulate the management of a particular item in a business process. The process components form a reusable library that you can call upon in different contexts. For example, you might have an item to "verify the caller's address/contact details" in the business process for taking out an insurance policy. This could be implemented as a sub-process and this particular process component could be used in the context of renewing an insurance policy.

Example

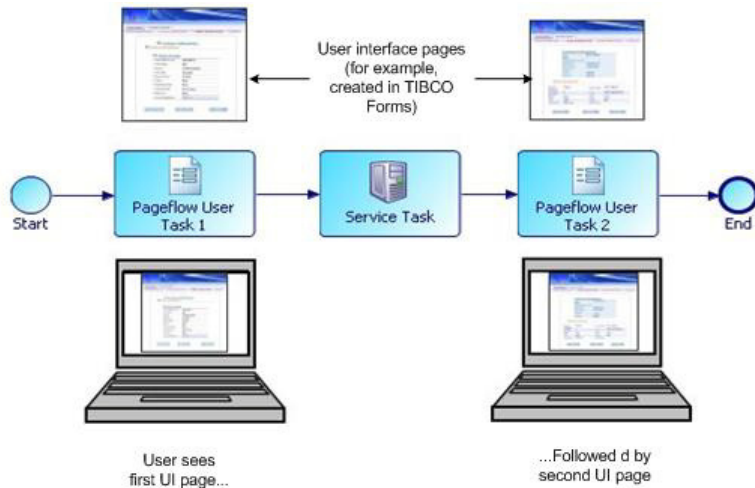
The following example shows a project and the associated packages and processes used in an insurance environment.



In this example, the Validate Policy process might call a sub-process in another package (for example, the Search Claim History process). This sub-process is in the same project in this example, but it could be located in a different project.

Pageflow Process Modeling

A *pageflow process* is a short-lived process designed to present user interface pages to the user in sequence. They are always executed by one person (the person that initiates the process instance). For example:



In this example, the user is presented with the user interface page (in this case a form created using TIBCO Business Studio Forms) associated with pageflow user task one. When the form is submitted, the service task runs. When the service task is completed, and the user interface page associated with user task 2 is displayed. The user is not aware of the service task, and sees one form followed directly by the next one.

A user task in a pageflow process differs from a user task in a business process in several key respects:

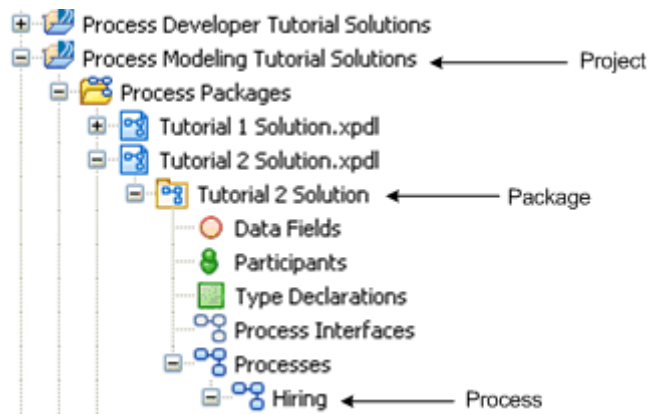
- Pageflow user tasks do not have participants assigned to them (this is because the user who initiates the process instance completes all the tasks in the pageflow process).
- Pageflow processes cannot contain lanes or pools.
- Pageflow user tasks do not create work items. The user interface pages are presented to the user without them needing to access their work queue.
- Pageflow user tasks do not restrict the type of technology used to create the user interface page that is displayed. For example, you could use TIBCO Business Studio Forms or a different technology. This allows the same process to be deployed to several runtime environments that utilize different user interface display technologies.

There are also special considerations to observe when using pageflows, specifically:

- Pageflow processes are not persistent (if the user cancels out of a pageflow process, data entered to that point is lost).
- Pageflow processes are not audited.
- Pageflow processes are not transactional (for example, there is no provision to roll back changes if a service task fails). If transactional control is required, chaining might be a better choice than a pageflow process (see [Refactoring Sub-Processes on page 197](#)).

About Projects, Packages and Processes

In TIBCO Business Studio there is a hierarchy consisting of project, package, and process:



Projects

TIBCO Business Studio supports the full project life cycle, bringing together all artifacts in a single place. The *project* is the container for these artifacts. As such, projects help to facilitate sharing and organization of resources. For example, team members may have different responsibilities but need to use the same resources that are made available through the TIBCO Business Studio project.

You must create a project to use TIBCO Business Studio. Each project has a corresponding directory in the file system (specified when you create the project). Projects can also refer to other projects (see [Creating Project References on page 52](#)).

Packages

A *package* is a mandatory container for processes and their infrastructure (such as *participants* and *data fields*). The package and any processes stored in it are saved in XPDL format. For example, in the insurance environment, separate packages could contain the processes used by the Claims Department, the Policy Origination/Maintenance Team and the IT Department. Processes can be shared between packages and projects so libraries of process components can be created and reused. For more information on reuse, see [Using Sub-Processes and Process Interfaces on page 189](#).



If the *Data Fields* folder is empty, it will be hidden by default. This is because the preferred usage is to define Data Fields at the Process level.

Processes

There are two types of process in TIBCO Business Studio:

- A *business process* models actual and future processes in your organization that usually involve more than one person. Business processes are short or long-lived.
- A *pageflow process* is a short-lived process (always executed in a single sitting) designed to implement a user interface dialog. It is always executed by one person (the person that initiates the process instance).

The Process Editor provides tools on a palette that use Business Process Modeling Notation (BPMN). By creating your process this way, you can fully prepare it for implementation by a specialist in your organization.

Some objects such as *business assets* can be shared at the project level. Others such as *data fields* and *participants* can be created at either the package level (where they can be shared amongst processes in that package), or at the individual process level (where they can only be used by that process).



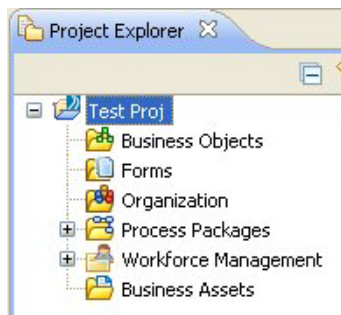
If the *Participants* folder is empty, it will be hidden by default. This is because the preferred usage is to define Participants at the Package level.

Process Interface

A *process interface* provides a way of allowing the dynamic selection of sub-processes at runtime. A process interface specifies the events and their parameters that must be present in processes created using that interface. At runtime, any of the processes that implement the interface may be chosen based on data available at that time. For more information, see [Process Interfaces on page 218](#).

Asset Types

Besides business process assets, there are other types of assets that you can include in your project. When you create the project, you can decide which types of assets to include, and also designate a special folder for each asset type. For example, the default analysis project has special folders for business objects, forms, organization models, process packages, and business assets:



For more information about special folders, see [Working With Special Folders on page 58](#).

Standards Support

TIBCO Business Studio supports the industry standards [Business Process Modeling Notation \(BPMN\)](#) and [XML Process Definition Language \(XPDL\)](#).

Business Process Modeling Notation (BPMN)



BPMN is a graphical notation, developed by the Business Process Management Initiative (BPMI) and now part of the Object Management Group (OMG), for representing the steps and flow of business procedures. The TIBCO Business Studio [Process Editor](#) supports BPMN 1.2.

For more information, see <http://www.bpmn.org>.

XML Process Definition Language (XPDL)



XPDL is used to represent the underlying structure of a business process to TIBCO Business Studio. Packages are stored in XPDL 2.1 format. Normally, you do not use XPDL directly, but indirectly by creating a process package then editing a process within it using the [Process Editor](#).

For more information, see <http://www.wfmc.org/xpdl.html>.

Capabilities and Perspectives

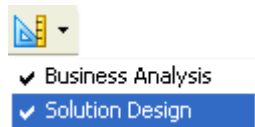
In Eclipse, a *perspective* includes the views and set of editors that you commonly use for a specific type of work. TIBCO has created several TIBCO Business Studio perspectives that include the views and editors you commonly use when creating business processes. A *capability* in Eclipse is a mechanism to enable and disable specific areas of the user interface (UI).

This guide describes the features of the Modeling perspective, which has the following two capabilities:

- **Business Analysis** This displays the areas of the UI relevant to a business analyst modeling processes using BPMN.
- **Solution Design** This displays the areas of the UI related to implementing a process. These features are described in destination-specific implementation guides.



If you are using TIBCO Business Studio, you can switch between these capabilities using the following menu:

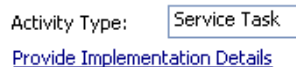


However, if you selected the installation profile Business Analyst, the Solution Design features are not available and selecting the Solution Design capability has no effect.



Studio for Analysts provides controls in a simplified user interface (ribbon format). You cannot switch between perspectives or capabilities when using Studio for Analysts.

If you have installed the Solution Design features, you can also see parts of the UI relevant to process implementation by clicking the provided links in the Properties view. For example:



Chapter 2

Getting Started for Analysts

Studio for Analysts is a simplified version of TIBCO Business Studio useful for business analysts or those responsible for high level or abstract modeling. The ribbon-based toolbar provides a familiar, easy to use interface and allows the user to access all the artifacts from a single file.

Topics

- [Modelled Application Archive, page 16](#)
- [The TIBCO Business Studio Ribbon, page 18](#)

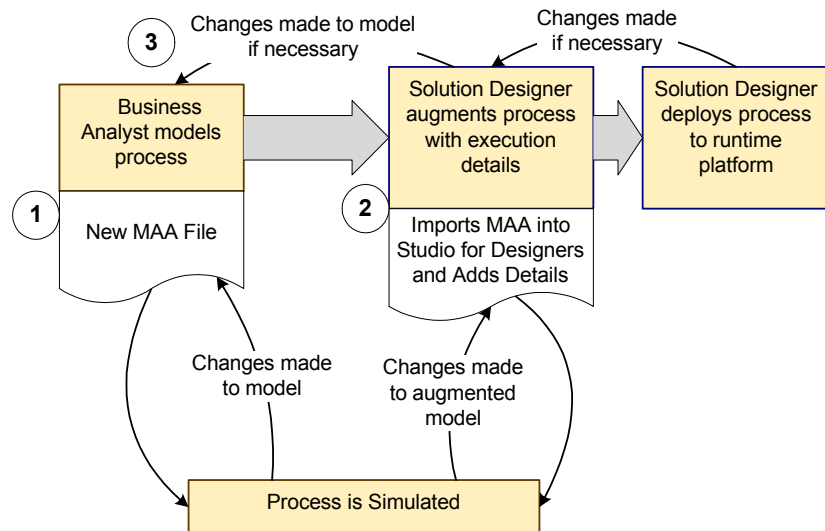
Modelled Application Archive

To enable the Business Analyst to access all the artifacts of a BPM project from a single location, TIBCO Business Studio saves the project as a Modelled Application Archive (MAA) file with a .maa file extension.

The MAA file contains information about the BPM project and its artifacts such as the organizational model, business object model, business processes, and packages.

Lifecycle of an MAA file

The following figure describes the lifecycle of a typical MAA file.



1. The Business Analyst models the process in TIBCO Business Studio for Analysts, and saves the project as an MAA file.
2. The Solution Designer imports the MAA into TIBCO Business Studio using the **Import > Existing MAA Files into Workspace** wizard.
3. If the model needs to be modified, the Solution Designer exports the updated MAA from the existing project and passes it to the Business Analyst. The Business Analyst makes the changes to the model and passes the updated MAA back to the Solution Designer. The Solution Designer then re-imports the MAA into TIBCO Business Studio.

Creating and Accessing MAAs

This section describes how MAAs are created and used.

Creating an MAA

To create a Modelled Application Archive (MAA), do one of the following:

- When you first launch Studio for Analysts, a dialog offering to create a new project, or open an existing project is displayed. Click the **Create a new project** link and enter the name of the new project in the **New Project** dialog.
- Select the New icon  from the ribbon and enter the name of the new project in the **New Project** dialog.
- For an existing BPM project, open the project in TIBCO Business Studio and export using the **Export > Export as MAA file** wizard. See [Exporting Projects to Modelled Application Archive \(MAA\), page 68](#) for details.



The derived artifacts such as generated WSDLs, business objects, forms, etc are not included in the generated MAA file and may need to be regenerated when you import the generated MAA file to a different workspace.

In all the above cases, the project is saved as an MAA file.

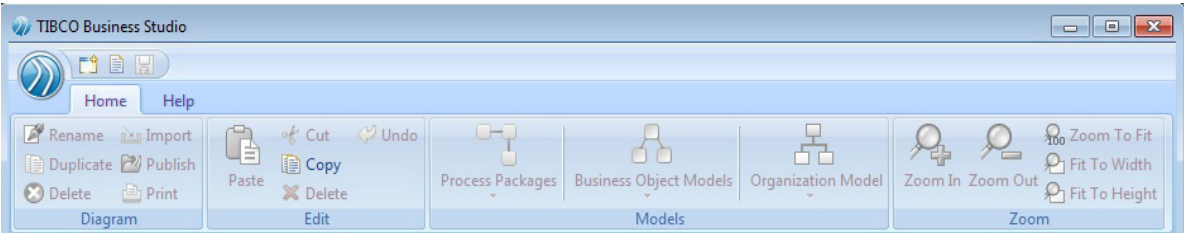
Accessing an MAA

To access the .maa file, do one of the following:

- Double-click the MAA file to open the project in Studio for Analysts.
- Click the Open icon  on the ribbon to look for an existing MAA file.
- You can open the MAA file in TIBCO Business Studio by using the **Import > General > Existing MAA files into Workspace** wizard. See [Importing Existing MAA Files into Workspace, page 65](#) for details.

The TIBCO Business Studio Ribbon

TIBCO Business Studio provides controls in the ribbon format. Also, the related commands are organized into groups which are provided in the horizontal bar at the top of the application window.



Following are the controls available at the top left-hand end of the ribbon.

Control	Description
New	Creates a new BPM project. When you click , the New BPM Project dialog appears with the following field: Name: Enter the name of the new project.
Open	Opens an existing BPM project. When you click , the Select File to Open dialog appears. • Select a .maa file - opens the BPM project with the Overview tab as the active tab. • Select an individual model (or process package) - opens the selected model in the edit mode. However, the other models are disabled. For example, if you open a process package (.xpd1 file), the Business Object Models and Organizational Models will be disabled.
Save	Saves the BPM project as an MAA file. When you save a new project for the first time, the Save New Project dialog allows you to choose the location of the MAA file. Subsequent changes to the project are saved to the file at the same location.

The **Home** menu provides access to the following groups:

Control	Description
Diagram	The controls in this group affect the selected model or package.
Edit	The controls in this group affect the selected object.
Models	The Models group consists of controls to create or edit Process Packages , Business Object Models , and Organizational Model . Working with Projects, Packages, and Processes describes these controls in detail.
Zoom	The controls in this group allow you to change the zoom level of the diagrams.

The following table describes the controls available within the group **Diagram**.

Control	Description
Rename	Renames the selected model or package. The Rename dialog allows you to enter the new name for the selected model or package.
Duplicate	Creates a copy of the selected model or package and appends the text <code>copy_</code> to the original name.
Delete	Deletes the selected model or package.
Import	Imports a model or package to the BPM project. See Importing and Exporting Projects for details.
Publish	Generates the documentation for all the assets in the BPM project and saves it in the selected destination folder. If the Open the documentation on completion checkbox is selected, the generated documentation is opened in the default browser.
Print	Prints the selected model.

The **Help** menu provides access to the following:

Control	Description
Help Contents	Opens the TIBCO Business Studio Help system.
About	Opens the About dialog which provides information about the TIBCO Business Studio version, features and plug-ins, and configuration details.

Control	Description
Configuration Details	Opens a dialog to provide the configuration details. You can also access the configuration details from Help > About > Configuration Details .
Preferences	Opens the Preferences page which captures the user information and preferred settings for the project.

Chapter 3

Getting Started for Designers

If you are new to using this product, there are several resources available to you from the Welcome page that is displayed when you first start TIBCO Business Studio. The Welcome page is also available from the **Help** menu. These resources include:

- Overview of the features available to you.
- Sample processes that you can install into your workspace.
- Tutorials that explain how to create a simple process. These tutorials will help you become familiar with the Process Editor.
- Links to the TIBCO Business Studio discussion forum and to the TIBCO Business Studio Developer Network.

Topics

- [Creating a Project, Package and Process, page 22](#)
- [Working with Processes, page 27](#)

Creating a Project, Package and Process



- If you have processes that you created using previous versions of TIBCO Business Studio, you must migrate them to the current version as described in [Migrating Processes Created in Previous Versions on page 36](#).
- This section describes how to create a project, and at the same time create a package and process using a template. You can however, perform each operation separately.

To create a new TIBCO Business Studio project, package and process, do the following:

1. Select **File > New > Analysis Project**.
2. Complete the **New Analysis Project** dialog as follows:

Field	Description
Project name	Enter a descriptive name for the project.
Location	Either accept the default location for the project (your workspace) or de-select the Use default location checkbox and click Browse to select a different location.
Version	Either accept the default version (1.0.0.qualifier) or enter a version for the project in the standard Eclipse format: <code>major.minor.micro.qualifier</code> The specified version will be the default for project artifacts such as process packages and organization models, and can be used to track revisions to the project. The version can be changed later as described in Changing Project Lifecycle Settings on page 60.
Status	Package life cycle status for informational purposes. How or whether you use life cycle statuses is up to you, but they are typically used as follows: • Under revision - for packages in development • Under test - for packages in User Acceptance Testing (UAT) • Released - for packages in production

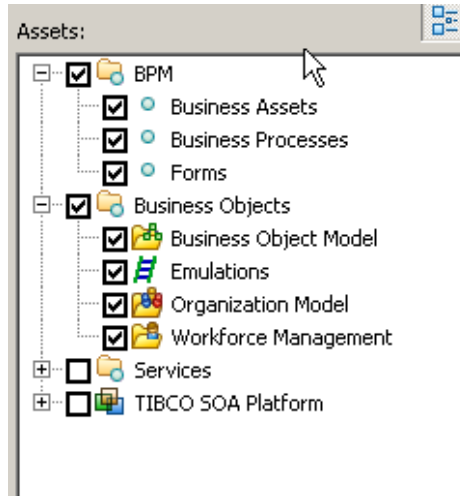
Field	Description
Destination Environments	Select the Destination Environment (optional). This specifies the intended runtime environment for project artifacts. For example, for processes, TIBCO Business Studio performs validation according to the selected destination environments. Other artifacts (for example, organization models) are not affected by project destination settings. For processes: • All processes are validated for BPMN 1.2 - additionally, other destinations can be selected to validate processes for use in specific environments • If you select BPM, the Process will be validated against BPM Version 1.0. • If you select Simulation, processes created under this project will be validated for simulation in TIBCO Business Studio as well as for BPMN correctness. • If you select iProcess, the Process will be validated against iProcess Engine Version 10.5 or higher. • If you do not select a destination environment, basic BPMN validation will be performed.



- To set a default destination which will then be ticked already in this dialog, select **Window > Preferences > User Profile**, and select the required destination from the dropdown next to the Project Destination field.
- The exact destination environments that are displayed depend on the edition of TIBCO Business Studio and any addins that you have installed.
The specific destination components that make up a destination environment can be viewed by selecting **Window > Preferences**, and selecting **Destination Environments**.
- To avoid error messages and warnings associated with modeling constructs that cannot be executed in the runtime environment, set the appropriate destination environment on the process.
- You can change or select the destination environment after the Process has been completed on the **Destinations** tab of the [Process Properties](#).

Click **Finish** on this dialog (or on any of the subsequent dialogs) to create a project with the settings you have made to that point. Click **Next** to modify the default project settings and create a process using a template.

3. The **Asset Type Selection** dialog displays the types of assets that you can include in your project.



For a full description of the different types of assets, see [Working with Assets on page 54](#). Select the types of assets you want to include in your project and click **Finish** if you are done, or **Next** to specify more options.

4. In the **Business Assets** dialog, enter the name of the folder that you want to designate for business assets files. Click **Finish** if you are done, or **Next** to specify more options.
5. In the **Select quality process project** dialog, select the name of the quality process project that you want to use. If you have not already created a Quality Process project in your workspace, you will receive a message telling you that there are no Quality Process projects in the current workspace, but you can add them later using the New Quality Process wizard. Click **Finish** if you are done, or **Next** to specify more options.
6. In the **Organization Model** dialog, enter the name of the folder that you want to designate for the organization model.

Select the checkbox under Model Details if you want to create an initial organization model.

Give the Organization Model you want to create a filename, or use the default one provided. The filename will automatically have a prefix of the domain name, as set in **Window > Preferences > User Profile**, and will end with *.om*.

Select the Default Schema Types checkbox unless you want to define different schema types.

You can also select to apply the default organization type to the Organization model you are creating.

Click **Finish** if you are done, or **Next** to specify more options.

7. In the **Business Object Model** dialog, enter the name of the folder that you want to designate for the Business Object Model.

Select the checkbox under Model Details if you want to create an initial Business Object Model.

Give the Business Object Model you want to create a filename or use the default one provided. The filename will automatically have a prefix of the domain name, as set in **Window > Preferences > User Profile**, and will end with *.bom*.

Select a Type for the Business Object Model. 3 types are provided for you to select from: the Business Object Model (no profile), Concept Model or Persistent Business Object Model.

Click **Finish** if you are done, or **Next** to specify more options.

8. In the **Emulations** dialog, enter the name of the folder that you want to designate for Emulations.

Click **Finish** if you are done, or **Next** to specify more options.

9. In the **Business Processes** dialog, enter the name of the folder that you want to designate for Business Processes.

Under Package Details, you can accept the pre-selected checkbox and either enter a filename or accept the default filename.

By default when you create a project, TIBCO Business Studio creates a package and process as well. The default packages Folder is called **Process Packages** and the default packages file is **ProcessPackage.xpdl**. Either accept the default names or rename the packages file and folder. Click **Finish** if you are done, or **Next** to specify more options.

10. The **Package Information** dialog is displayed. Either accept the default properties of the package, or modify them as necessary and click **Finish** if you are done, or **Next** to specify more options.

- **Package Label** Descriptive label for the package. Defaults to the same name as the filename of the package and is used for purposes such as simulation reports.

- **Author** Username of the user who created the package.



The name of the author can be defined in the User Name: field on **Window > Preferences > User Profile**. If no user is defined there, it uses the default user system property.

- **Created** Displays the date/time that the package was created.
- **Description** Text description of the package.
- **Documentation Location** URL or filename of any supporting documentation.
- **Status** Package life cycle status for informational purposes. How or whether you use life cycle statuses is up to you, but they are typically used as follows:
 - Under revision - for packages in development
 - Under test - for packages in User Acceptance Testing (UAT)
 - Released - for packages in production
- **Business Version** Version information about the package (this is inherited from the setting in the **New Analysis Project** dialog in [step 2](#)).



The version you specify in this field is unrelated to any source control system and their use is entirely user-defined.

- **Cost Unit** Three-letter currency code that provides context for the costs reported in Simulation. For example, there is a Cost Per Unit for Participants in Simulation, that displays an integer with no currency. For example, use this informational field to specify to anyone using the process that a Cost Per Unit of 50 is intended to be read 50 USD. GBP is the default.



Both **Cost Unit** and **Language** provide content assist. Press Ctrl+Space to see a list of allowed values for these fields.

- **Language** Provides context for user-visible language in processes. For example, annotations in a process may be in a language or character set unfamiliar to the user of the process. Use this field to specify the language used. Note that this field is informational; it does not change any system or TIBCO Business Studio-specific language settings.
11. The **Select Template** dialog allows you to create a process using a template. Select a template from those available and click **Finish** if you are done, or **Next** to specify more options.
 12. The **Set Special Folders** dialog displays the default special folders for each asset type that you selected in the previous step. Either accept the default names or enter your own names and click **Finish**.

The newly-created package, process, and project are displayed in the Project Explorer. For more information about the Project Explorer, see [Project Explorer View on page 250](#).

Working with Processes

This section describes some common tasks and shortcuts to help you use the Process Editor to create processes. For additional information, see subsequent chapters and [Tips and Tricks on page 271](#).

Adding Objects

You can add objects either using the gadgets that appear on an existing object or using the tools on the palette.

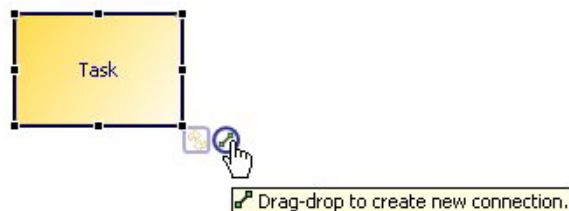
Using an Object's Gadget to add objects

A gadget is a user interface aid that allows you to easily create sequence flows or other links between objects.

The advantage of using gadgets is that it is quick to do, and can be used to perform many, but not all, of the tasks traditionally performed from the palette.

When an object is selected, initially one gadget of each applicable type is displayed. If multiple gadgets of a particular type are available, these are displayed when the mouse cursor is moved over the first.

For example, if a task is selected, the **Create Connection** gadget is visible to the lower right of the task:



Gadgets that can be used to designate references to other objects can be used to designate a reference (drag-drop gadget onto target object) or go-to a currently referenced object.

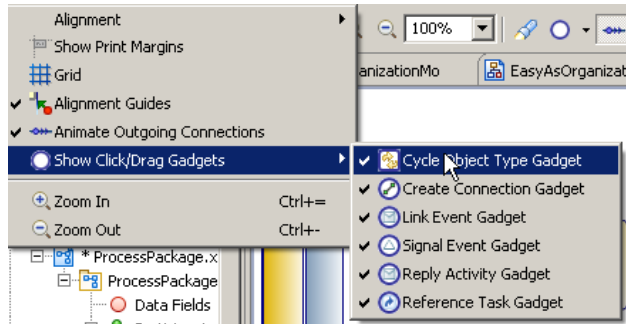
The following gadgets are provided:

-  **Cycle Object Type Gadget** Change the activity type by cycling through the different types.
-  **Create Connection Gadget** Create sequence flows to existing object or create and connect to new object.
-  **Link Event Gadget** Designate a throw and catch link event pair.

-  **Signal Event Gadget** Designate a throw and catch signal event pair.
-  **Reply Activity Gadget** Specify a request - reply relationship (with throw and catch message events or send and receive tasks).
-  **Reference Task Gadget** Select the task that the current task will reference.



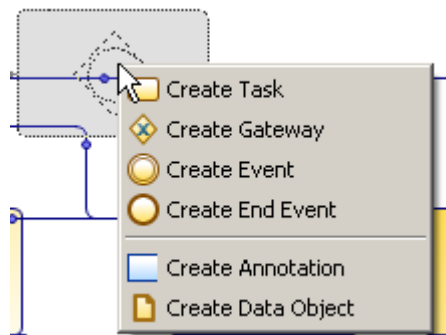
You can control the display of gadgets using the toolbar tool, or the **Show Click/Drag Gadgets** drop-down list from the **Diagram** toolbar:



To add an object, click the **Create Connection** gadget, and drag and drop it in a lane content area or embedded sub-process content area.

A popup menu is displayed showing a selection of applicable main object types that can be created (available types governed by the source object and drop location).

You can also drag onto an existing sequence flow, and the object is inserted onto the sequence flow. If the target flow is not to or from the selected object then a new flow added to the new object.

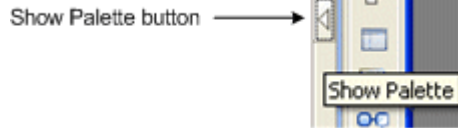


Using the Palette to add Objects

Add an object (for example, a task) by doing the following:



When you first start TIBCO Business Studio, if the palette is not visible, expand it by clicking the **Show Palette** button to the right of the window:



1. Click the appropriate tool in the palette. If an object has more than one type (for example, a task can be a service task, or user task, and so on), each type has a tool on the palette. For example:



Click another tool to display the available sub-types. For example, click the Gateway tool to see the different types of gateways; this closes the currently open tool drawer if more space is required to open the new tool drawer. To change this behavior and cause the available tasks to remain open, click the **Pin Open** button ().



- To add more than one object of the same type, hold down the **Ctrl** key.
- You can customize certain palette preferences by selecting **Customize** from the right-click menu on the palette. This allows you to define preferences such as the default tool drawer states on editor startup. There is also a Favorites drawer that you can customize (see [Customizing Palette Favorites on page 33](#)).

- Position the pointer on the process at the place where you want to place the object. The pointer changes shape: 
- Click the process to add the object.



You can also drag and drop from the palette. So select an item on the palette and drag and drop it to where you require it in the process.

Connecting Objects

You can connect objects either using the gadgets that appear on the object or using the connection tools on the palette.

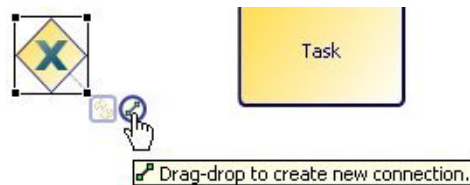
Using an Object's Gadget to connect objects

A gadget is a user interface aid that allows you to easily create sequence flows or other links between objects.

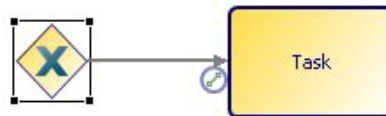
See [Using an Object's Gadget to add objects, page 27](#) for more information on the different gadgets available.

Connect objects in the process as follows. The example in this section shows creating sequence flow using the **Create Connection** gadget, but a similar procedure applies to all connection types:

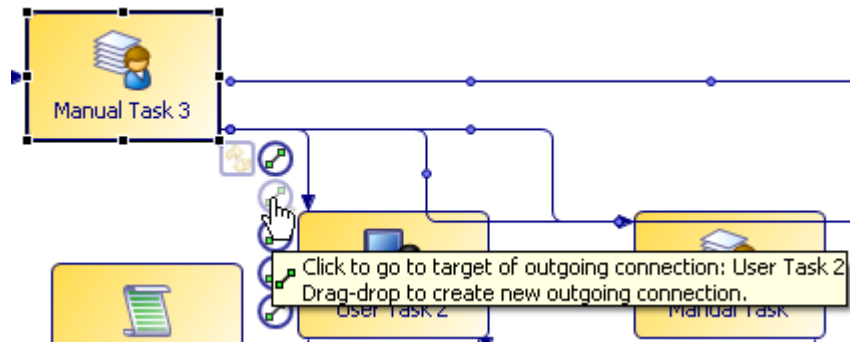
- Select the object. For example:



- Drag the **Create Connection** gadget to connect to another object. For example:



3. If you add more connections or references, additional gadgets are added. These are only visible if you position the mouse over the visible gadget. For example:



In this example, there is a **Create Connection** gadget for each outgoing sequence flow. Clicking a gadget selects the referenced or connected object in the Process Editor (for example, clicking the gadget attached to a reference task selects the referenced task).

Using Connection Tools on the Palette

To connect two objects with sequence flow, do the following:

1. Click the **Sequence Flow** tool.
2. Position the pointer over the object that will be the source of the sequence flow.
The pointer changes shape: 
3. Draw the sequence flow using one of the following methods:
 - Click the source object, then position the pointer over the target of the sequence flow. If you are satisfied with the shape of the flow, click the target.
 - Click the source object, holding down the mouse button, drag the pointer to the target object and release the mouse button.



While drawing the sequence flow, you can add bendpoints by right-clicking or pressing the spacebar as you create the flow.



You can also drag and drop from the palette. So select an item on the palette and drag and drop it to where you require it in the process.

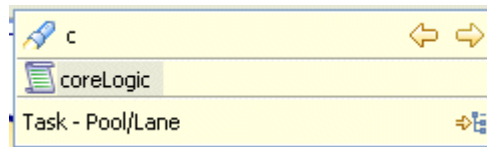
Selecting Objects

Select objects using one of the following methods:

- Using the **Select** tool on the palette. Hold down the **Ctrl** to add single objects to the selection or **Shift** keys to select a range of objects.
- Using the **Marquee** tool. This allows you to drag a rectangle around all the objects that you want to select.

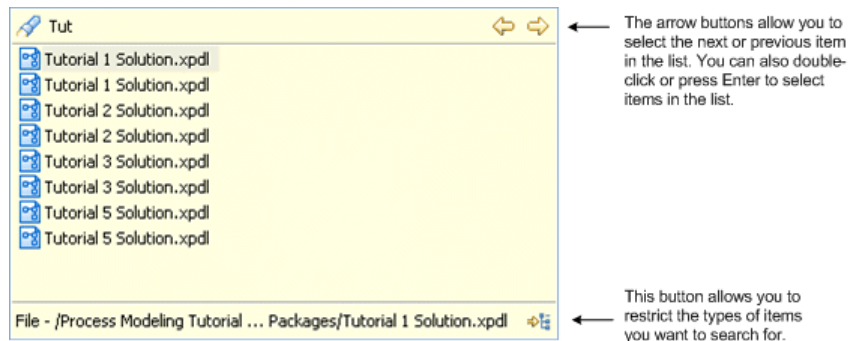
Finding Objects

In small processes, it is easy to see all the objects in the diagram. However, in large processes it can sometimes be difficult to locate objects. To find an object on a process, press **Ctrl+F** or click the  icon on the toolbar. In the resulting dialog enter the name or partial name of the object you are looking for. For example:



In this example the only task that starts with "c" is a script task called *coreLogic*. Clicking on the task displays it in the Process Editor.

You can also use the search facility to locate workspace elements such as processes, package names, and process interfaces by pressing **Ctrl+F** or the search toolbar button in the Project Explorer view. For example:



You can also search for items using the first characters from the start of each capitalized word (CamelCase searching). For example, "BP" returns a file called **BankPayment.wSDL**.

The wildcard ? returns all elements. Use the * wildcard to restrict the results (for example, *2 to return all matches ending in 2). Note that the wildcard * by itself does not return any results; it only works in conjunction with a string.

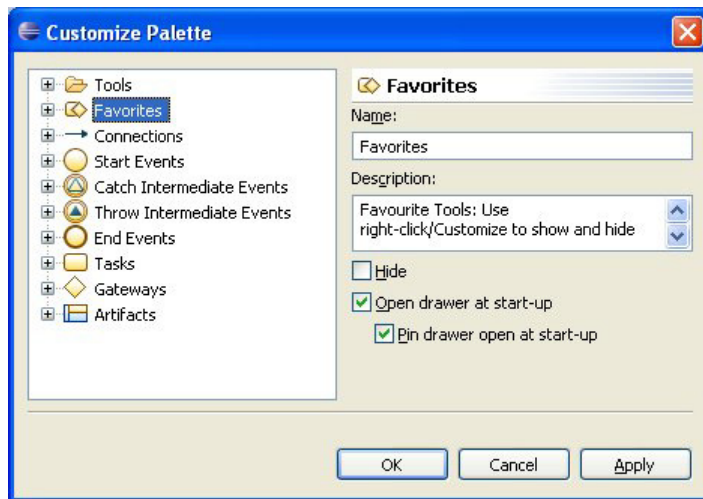
Customizing Palette Favorites

The palette contains a drawer labelled **Favorites** that contains some of the most commonly used objects:



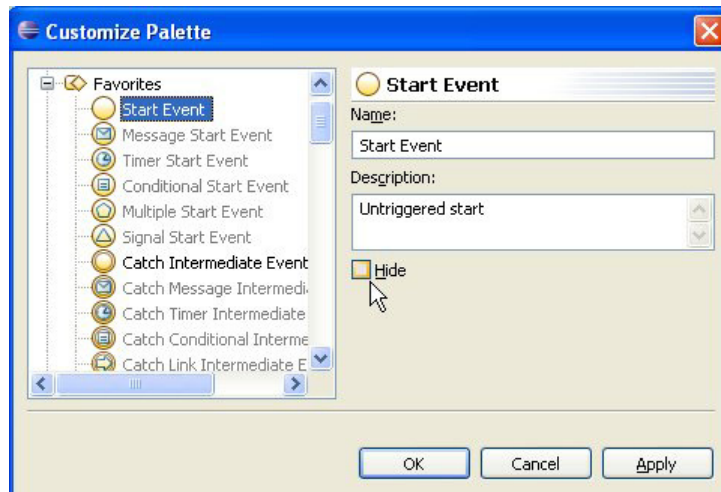
You can add or remove objects from the Favorites drawer by doing the following:

1. Right-click in the Favorites drawer, and select **Customize**.



2. Expand the **Favorites** item in the **Customize Palette** dialog.

3. Select an object and use the **Hide** check box to display or hide the object in the favorites drawer:



Chapter 4

Working with Projects, Packages, and Processes

This chapter describes how to work with projects, packages, and processes in TIBCO Business Studio. For an overview, see [About Projects, Packages and Processes on page 9](#).

Topics

- [Migrating Processes Created in Previous Versions, page 36](#)
- [Creating a Package, page 40](#)
- [Copying a Package, page 41](#)
- [Creating a Business Process, page 42](#)
- [Creating a Pageflow Process, page 46](#)
- [Deleting a Process, page 49](#)
- [Creating Extended Attributes, page 51](#)
- [Creating Project References, page 52](#)
- [Working with Assets, page 54](#)
- [Working With Special Folders, page 58](#)
- [Managing Project Lifecycle, page 60](#)
- [Importing and Exporting Projects, page 63](#)
- [Using Process Fragments, page 69](#)
- [Using Source Control, page 72](#)
- [Comparing Different Versions of a Process Package on page 75](#)
- [Emailing Processes, page 80](#)
- [Creating Documentation, page 81](#)
- [Using Presentation Channels, page 84](#)

Migrating Processes Created in Previous Versions

Processes created in previous versions of TIBCO Business Studio must be migrated to the current version before they can be opened. The migration process is different depending on which version of TIBCO Business Studio you are migrating from.

See [Importing and Exporting Projects on page 63](#) for information on migrating an entire project.

Migrating from Version 1.1

To migrate from Version 1.1, do the following:

1. Depending on whether you are migrating a workspace or a project, do the following:



TIBCO recommends that you create a backup copy of the current workspace before migrating.

- To migrate a workspace, select **File > Switch Workspace > Other...**, browse for the location of the Version 1.1 workspace, and when prompted, restart TIBCO Business Studio.
- If you want to migrate only one project, and not an entire workspace, either copy the project into the Project Explorer or select **File > Import**, expand **General**, select **Existing Projects into the Workspace**, and follow the instructions in the wizard to import the project.



Whilst the Project folders are present, the Project Explorer displays the contained XPD files only, without the standard TIBCO Business Studio Project Explorer branches such as Process Packages Folders, Data Fields, and so on.

2. Right-click a project you want to migrate and select **Migrate Project to Current Version**.

The project is migrated to the current version and you can open the process in the Process Editor.



If your project contains Business Assets, you must migrate the Business Assets separately. [Migrating Business Assets on page 37](#) explains how to migrate the Business Assets.

Migrating from Version 2.x or 3.x



TIBCO recommends that you run **Project > Clean...** soon after copying the package into the workspace. You can then use the **Quick Fix** wizard to resolve the problems listed in the Problems view.

1. Copy the package containing the process into a packages special folder (see [Enabling and Disabling Special Folders on page 59](#)).
2. Refresh the Project Explorer in TIBCO Business Studio.
3. Note that the package cannot be expanded and clicking on it results in an error which is listed in the Problems view for the package.
4. In the Problems view for the package, the following error message is displayed:

 Problems while loading the file. Possible version problem.

Right-click and select **Quick Fix**. When you have resolved the problems, you can open the process in the Process Editor.



If your project contains Business Assets, you must migrate the Business Assets separately. [Migrating Business Assets on page 37](#) explains how to migrate the Business Assets.



The following procedure lets you correct several migration problems at once from the **Quick Fix** dialog:

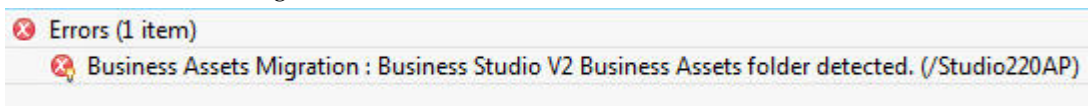
1. The Problems view displays all the error messages in the workspace.
2. Right-click an error message and select **Quick Fix**. The Quick Fix dialog lists all the resources to which the fix can be applied. Select the required resources and then click **Finish**.

The number of problems listed depends on the filtering that you have set up in the Problems view (see [Problems View on page 275](#)).

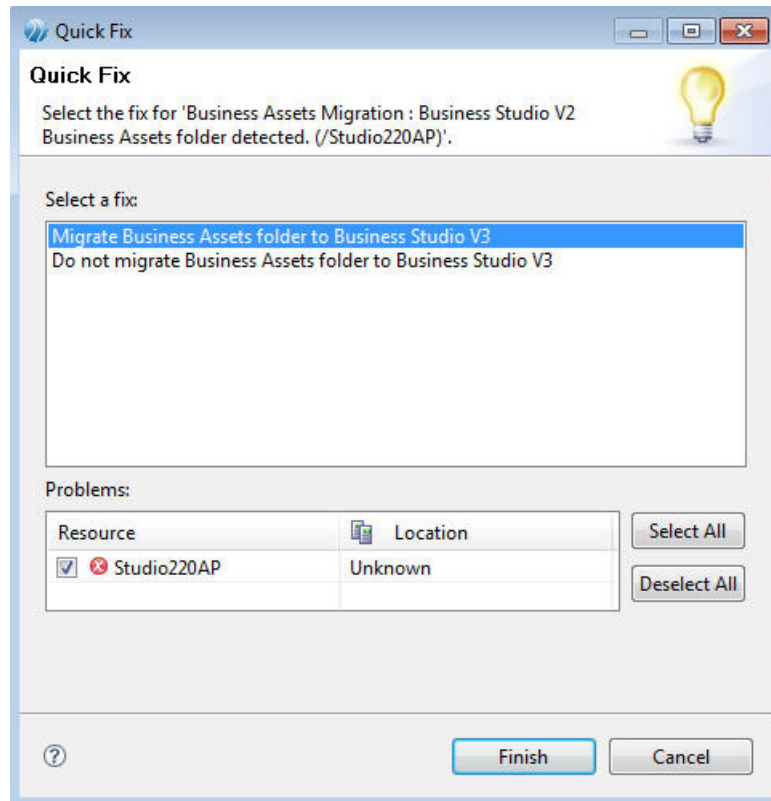
Migrating Business Assets

Business assets from a previous version of TIBCO Business Studio cause an error in the Problems view. Use the following procedure to migrate business assets created in previous versions of TIBCO Business Studio to the current version.

1. Copy the folder and contents of the project containing the business assets into your workspace.
2. Select **File > Import**, and from the resulting dialog, select **General > Existing Projects into Workspace** and click **Next**.
3. Click **Browse** and select your Eclipse workspace as the root directory for the project import. The specified workspace is scanned for potential projects for import.
4. Select the checkbox for the project you want to import and click **Finish**.
5. The newly imported project is displayed in the Project Explorer, as well as the following error in the Problems view:



6. Right-click the problem and select **Quick Fix**. The following dialog is displayed:



7. Select one of the options and click **OK**:
 - **Migrate Business Assets folder to Business Studio V3** Select this option to clear the problem from the Problems view and migrate the business assets to Version 3.x. This changes the **Business Assets** folder to a special folder.
 - **Do not migrate Business Assets folder to Business Studio V3** Select this option to clear the problem from the Problems view and does not change the Business Assets folder to a special folder. You can however manually change the Business Assets folder to a special folder as described in [Working With Special Folders on page 58](#).

In either case, the problem is removed from the Problems view.

Creating a Package

To create a package, do the following:



You can create a package and process as part of creating a project (see [Creating a Project, Package and Process on page 14](#)).

1. Right-click the Process Package folder under which you want to create the package and select **New > Process Package**.
2. If you want to change the Packages Folder under which the package is created, click **Browse** or enter an existing folder name.
3. Enter the name of the **File** or accept the default filename. The filename must have **.xpdI** as the extension. Click **Next**.
4. The **Package Information** dialog shows the name of the package and its default properties (for a full explanation, see [step 10 in Creating a Project, Package and Process on page 14](#)).

Either accept the default properties of the package, or modify them as necessary. Then, click **Finish** to create a process using the default template, or click **Next** to create a process using a different template.

5. By default a process will be created with start and end event connected by sequence flow. The **Select Template** dialog allows you to create a process using a different template. Select a template from those available and click **Finish**.

The newly-created packages and any processes that you elected to create are displayed in the Project Explorer.



To avoid error messages and warnings associated with modeling constructs that cannot be executed in the runtime environment, set the appropriate destination environment on the process (see [Destinations and Validation on page 236](#)).

Copying a Package

You can copy a package into the same project, or into a different project in your workspace as follows:

1. In the Project Explorer, highlight the package file you want to copy.
2. Press **Ctrl+C** to copy the package.
3. Open the project into which you want to copy the package.
4. Select the folder into which you want to copy the package (for example, the **Process Packages** folder).
5. Press **Ctrl+V** to paste the package. If copying the package within the same project, you are prompted to enter a new name. If copying the package to a different project when a package of that name already exists, you are prompted to overwrite the file.



After copying a package, errors similar to the following are displayed in the Problems view:

✖ XPDL 2.0 : The following files have duplicate process IDs: /mond/Process Packages/ProcessPackage.xpdl

To correct this, right-click the problem, and select **Quick Fix**. Accept the suggested fix for this problem to resolve the duplicate IDs.

Creating a Business Process

To create a new business process, do the following:



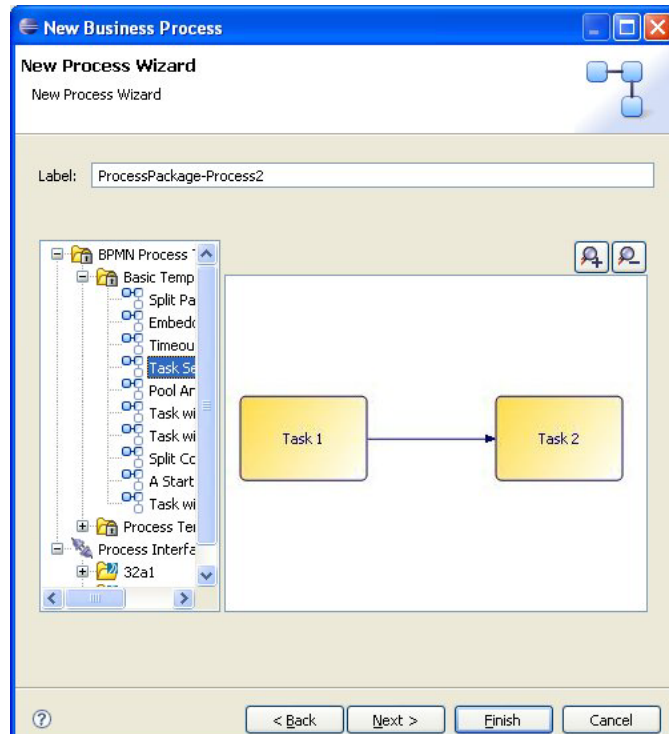
To create a Process and its containing package and Project in one operation, see [Creating a Project, Package and Process on page 14](#).

1. Before creating a process, you must create a project and a package to contain your process.
2. In the Project Explorer, select the package you created, right-click and select **New > Business Process**.
3. The **New Business Process** wizard is displayed.



If you start this wizard from the **File > New** menu, the first dialog is the **Project and Package** dialog, where you must specify a valid project and package. This dialog is not displayed if you right-click at the package level to start the dialog; however you can click **Back** to display it if necessary.

4. Enter the **Label** of the process. To use a template to create the process, select the template and click **Next**. The following example shows one of the basic templates (a task sequence).



In addition to the process templates, you can select a process interface as the basis for your new process. This creates a process with the necessary events, parameters that are specified in the process interface.

5. In the **Description** dialog, add optional text that describes the process, an optional URL that links to documentation about the process, and click **Next**.
6. In the **Destinations** dialog, select the **Destination Environment** (optional). This controls the validation that TIBCO Business Studio performs when you save the process:



- The exact destination environments that are displayed depend on the edition of TIBCO Business Studio that you have installed.
- The specific destination components that make up a destination environment can be viewed by selecting **Window > Preferences**, and selecting **Destinations**.

- All processes are validated for BPMN 1.2 - additionally, other destinations can be selected to validate processes for use in specific environments
- If you select **BPM**, the Process will be validated against BPM Version 1.0.
- If you select **iProcess**, the process will be validated for direct deployment to the TIBCO iProcess Engine 10.5 or higher.
- If you select **Simulation**, the Process will be validated for simulation in TIBCO Business Studio as well as for BPMN correctness.
- If you select **Modeling**, the Process will be validated for modeling in TIBCO Business Studio as well as for BPMN correctness.

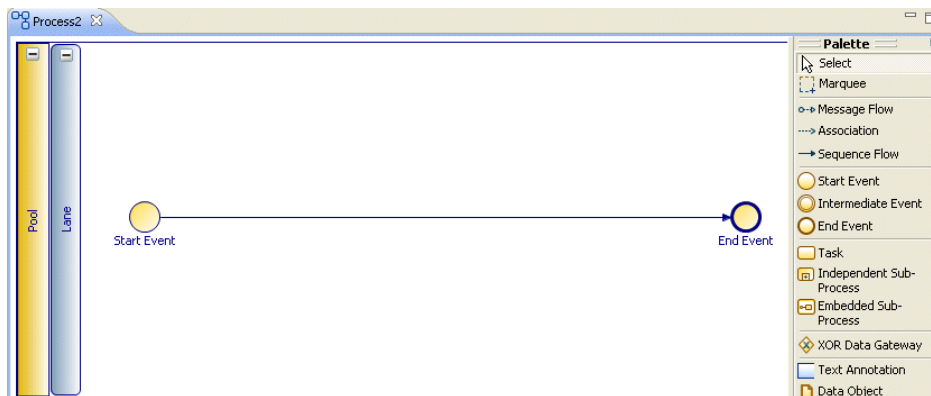
If you do not select a destination environment, basic BPMN validation will be performed.



To avoid error messages and warnings associated with modeling constructs that cannot be executed in the runtime environment, set the appropriate destination environment on the process.

You can change or select the destination environment after the Process has been completed on the **Destinations** tab of the [Process Properties](#).

7. The **Extended** dialog is displayed. This allows you to add optional supplemental information to the XPD L for the process. For more information see [Creating Extended Attributes on page 51](#).
8. Click **Finish**.
9. The process that you created is displayed in the Process Editor. When you first start the Process Editor, the palette (on the right side of the diagram) might be collapsed; if so, expand it:



You can expand this window to fill your screen by double-clicking the title bar (in this example, the area labeled **Process2**).

The Process Editor provides a **Pool** and **Lane** that you can rename if you plan on using pools and lanes (see [Swimlanes on page 121](#)). You add elements of your business process using the [Palette](#).

Creating a Pageflow Process

A pageflow process is a short-lived process designed to display user interface pages to the user. All tasks that are available in a business process are available within a pageflow process with the exception of a *business process user task*. Pageflow processes have a special variant of a business process user task that does not have participants, and does not generate work items. These are referred to as *pageflow user tasks*.

A pageflow process is stored under the **Processes** branch of the Project Explorer alongside business processes.

To create a new pageflow process, do the following:



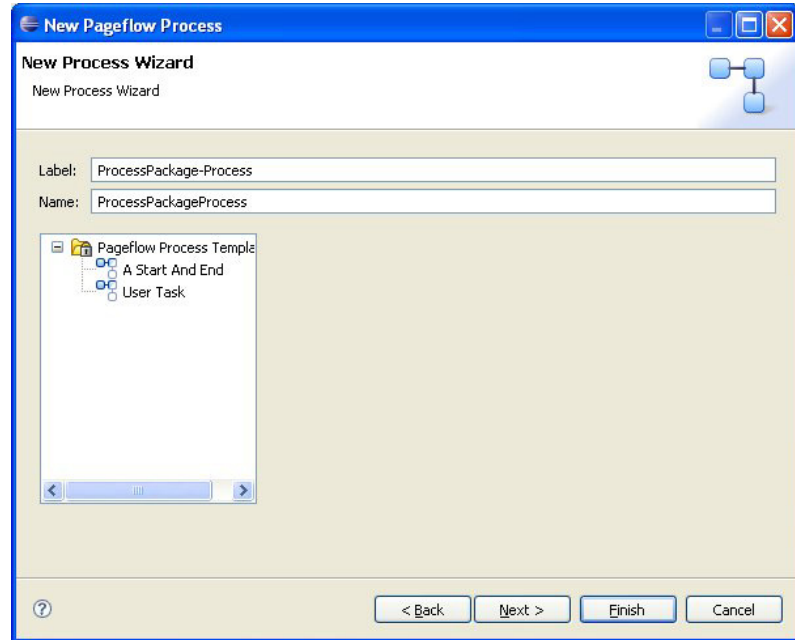
To create a Process and its containing package and Project in one operation, see [Creating a Project, Package and Process on page 14](#).

1. Before creating a pageflow process, you must create a project and a package to contain your process.
2. In the Project Explorer, select the package you created, right-click and select **New > Pageflow Process**.
3. The **New Pageflow Process** wizard is displayed.



If you start this wizard from the **File > New** menu, the first dialog is the **Project and Package** dialog, where you must specify a valid project and package. This dialog is not displayed if you right-click at the package level to start the dialog; however you can click **Back** to display it if necessary.

4. Enter the **Label** of the process. If you want to use a template to create the process, select the template and click **Next**.



5. In the **Description** dialog, add optional text that describes the process, an optional URL that links to documentation about the process, and click **Next**.



The **Documentation Url** field is intended for design-time collaboration; it is not displayed in the runtime environment.

6. In the **Destinations** dialog, select the **Destination Environment** (optional). This controls the validation that TIBCO Business Studio performs when you save the process:



- The exact destination environments that are displayed depend on the edition of TIBCO Business Studio that you have installed.
- The specific destination components that make up a destination environment can be viewed by selecting **Window > Preferences**, and selecting **Destinations**.

If you do not select a destination environment, basic BPMN validation will be performed.



To avoid error messages and warnings associated with modeling constructs that cannot be executed in the runtime environment, set the appropriate destination environment on the process.

You can change or select the destination environment after the Process has been completed on the **Destinations** tab of the [Process Properties](#).

7. The **Extended** dialog is displayed. This allows you to add optional supplemental information to the XPDL for the process. For more information see [Creating Extended Attributes on page 51](#).
8. Click **Finish**.
9. The process that you created is displayed in the Process Editor:



When you first start the Process Editor, the palette (on the right side of the diagram) might be collapsed; if so, expand it. You can expand this window to fill your screen by double-clicking the title bar. A pageflow process has a different default color scheme from a business process.

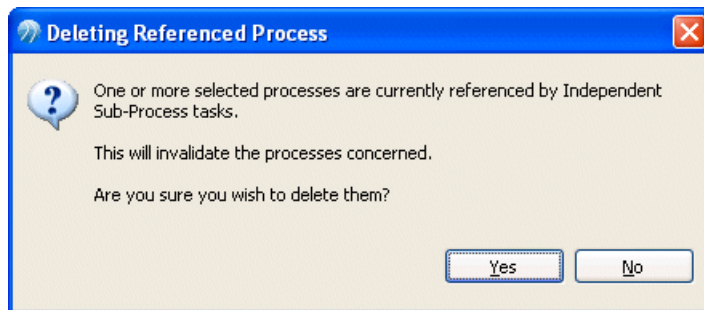


Pageflows do not contain pools or lanes.

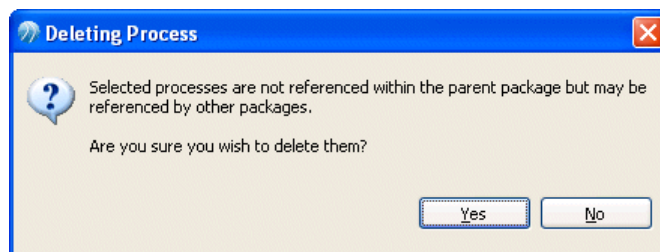
Deleting a Process

To delete a process:

1. Do one of the following:
 - Right-click the process in the Project Explorer, and select **Delete**.
 - Highlight the process and select **Edit > Delete**.
 - Highlight the process and press **Delete**:
2. If the process you are deleting is referenced in the parent package (for example, you are deleting a sub-process), the following dialog is displayed:



- If you are sure that you want to invalidate the process that is referring the the process you are deleting, click **Yes**.
 - If you do not want to delete the process, click **No**.
3. If the process you are deleting is not referenced in the parent package, it is still possible that it is referenced by a process in a different package. TIBCO Business Studio cannot validate against this however and the following dialog is displayed:



- If you are sure that processes in different packages do not reference the process you are deleting, click **Yes**.
- If you do not want to delete the process, click **No**.



It is also possible to delete a process by deleting the XPDL file of the package that contains the process in Windows Explorer. This will delete all processes in that package. If you delete a package in Windows Explorer while TIBCO Business Studio is open, you must refresh the Project Explorer to see an updated view of the workspace.

Creating Extended Attributes

Extended attributes can be used to add extra information to the schema such as:

- **XML** For example, an XSD file used by a web service.
- **Plain text** For example, Supply Chain Management (SCM) metrics and best practices.

To add extended attributes to an object, do the following:



You can also add extended attributes when creating a new Process or Process Interface as described in [Creating a Business Process on page 42](#), [Creating a Pageflow Process on page 46](#) and [Creating a Process Interface on page 221](#).

1. Click the **Extended** tab.



This tab is only available on the Properties view for objects that support extended attributes.

2. Click **Add** to add a new extended attribute.
3. Modify the following:
 - **Name** Provide a **Name** for the attribute (you cannot include spaces in the name).
 - **Value** Add any text for the **Value**.
 - **Escape Body** Select this check box if you want the parser to interpret the contents of the body attribute as text rather than XML (you do not want the parser to interpret symbols such as < or >). If you are entering XML, do not select this checkbox so the parser can validate the body attribute.
 - **Body** Enter whatever text or XML you want to make up the body of the attribute.
4. Save the package that contains the process.

Creating Project References

Projects can refer to other projects. This allows you to do the following:

- Invoke a sub-process from another project (see [Reusable Sub-Processes on page 190](#)).
- Use a process interface from another project (see [Process Interfaces on page 218](#)).
- Create a data field or formal parameter of the type external reference (see [Creating Data Fields and Parameters on page 164](#)).
- Select a WSDL from a different project.

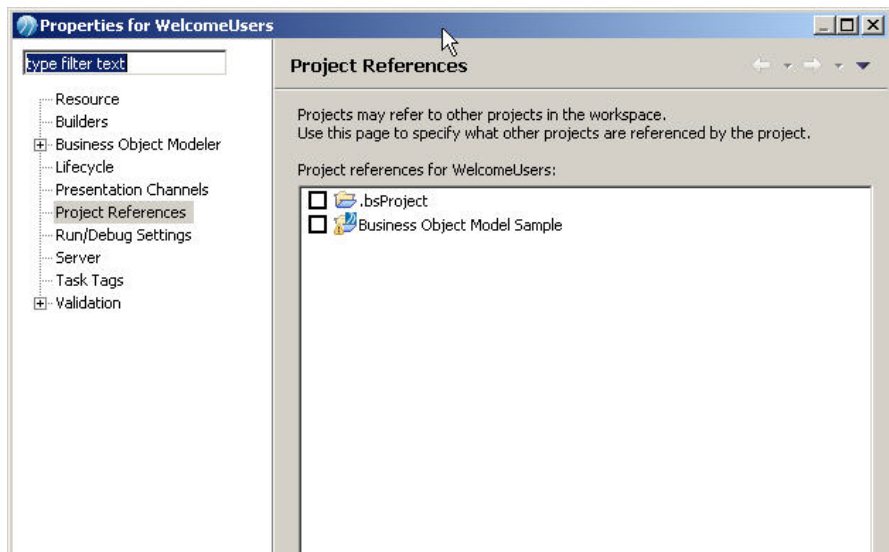
Project references can be created in two ways:

- Explicitly, in the project properties (see the following section).
- Automatically (subject to user confirmation), when you select an object from outside the current Project (see [Creating Project References in a Selection Dialog on page 53](#)).

Creating References in the Project Properties

To create a reference from one Project to another in the project properties, do the following:

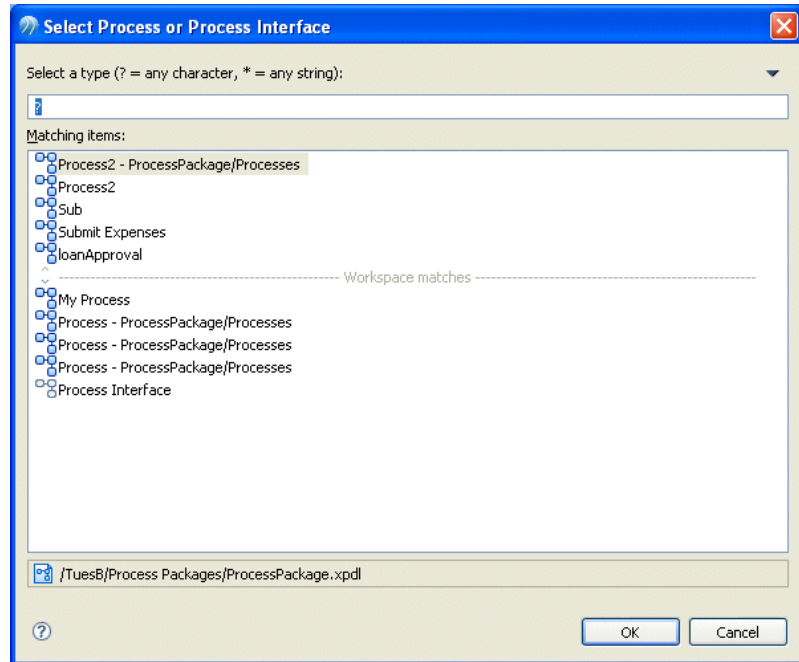
1. Right-click the project, select **Properties**, and highlight **Project References**:



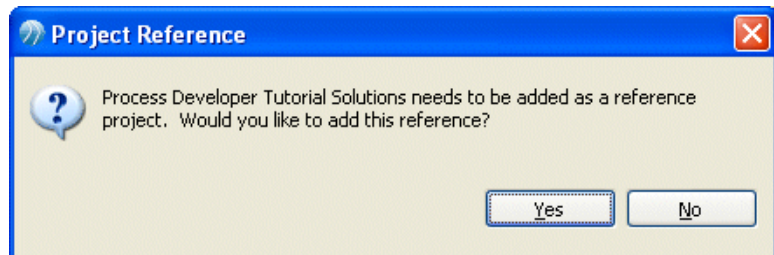
2. In this dialog, select any projects that you want to refer to from the selected project.

Creating Project References in a Selection Dialog

Project references can automatically be created when an object from outside the project is selected. For example, when defining a sub-process call, you select from a list of all the sub-processes in the workspace:



If the selected sub-process is not part of the project in which the sub-process call is located, the following message is displayed:



Clicking **Yes** creates a reference to the project that contains the sub-process.

Working with Assets

Assets include XPDL package files, WSDL files, documents, business object models, and so on that relate to the project. These are usually stored in special folders under the project and include the following:

- [Business Process Assets](#)
- [Forms Assets](#)
- [Organization Model Assets](#)
- [Service Assets](#)
- [Business Object Model Assets](#)
- [Business Assets](#)

Because TIBCO Business Studio is extensible, there may be other types of assets displayed. For more information about special folders, see [Working With Special Folders on page 58](#).

Business Process Assets

Business process assets include the XPDL package file and all the associated processes, data fields, parameters, and so on. The default special folder for business processes is called **Process Packages**.

Forms Assets

Forms assets include all the files and parameters necessary to display the forms that you have created in your project. These include XML, HTML, and properties files, as well as the parameters used by the forms. The default special folder for these assets is called **Forms**.

Organization Model Assets

Organization Model assets include the organization model and its components, such as organization units, positions, groups, and so on.

Service Assets

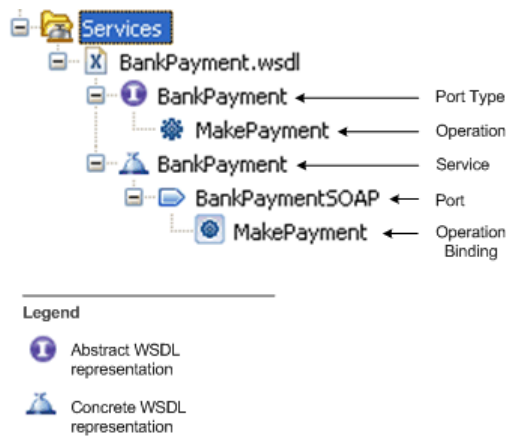
Service assets include the WSDL files for any web services that you import into your project. The default special folder for these assets is called **Service Descriptors**. You can specify either an abstract or concrete WSDL file:



An Analysis project does not have a service descriptors special folder by default. If you need to create a special folder, see [Working With Special Folders on page 58](#).

- An *abstract WSDL* document defines an abstract messaging model without reference to protocols or encodings, and consists of port types and operations.
- A *concrete WSDL* document contains the abstract definitions and the communication protocols and data formats by which the operations defined in the abstract WSDL document can be invoked. A concrete WSDL file consists of the elements from the abstract WSDL file as well as an operation binding.

In the Project Explorer, both the abstract and concrete parts of the WSDL file are represented when you expand the WSDL file:



Business Object Model Assets

Business object model assets include the files for any business object models that you create.

Business Assets

There are two categories of project-related business assets in TIBCO Business Studio:

- **Quality Process** Business cases, project plans, and so on.
- **Ad-hoc Assets** Supporting documents, spreadsheets, and so on that are not part of the quality process.

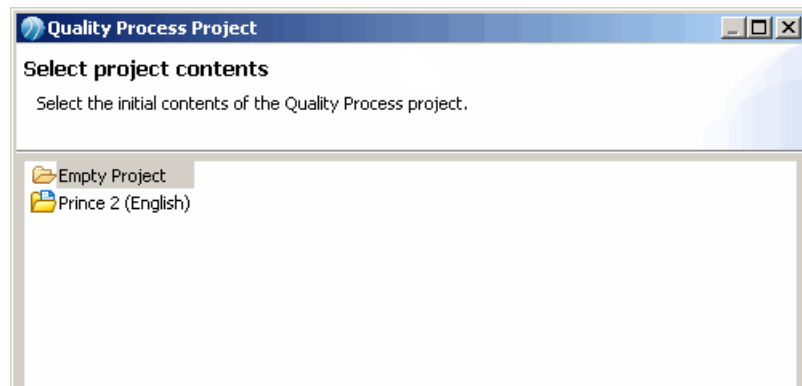
Quality Process Business Assets

TIBCO Business Studio provides assets to support quality processes or methodologies. To include quality process templates, you must do the following:

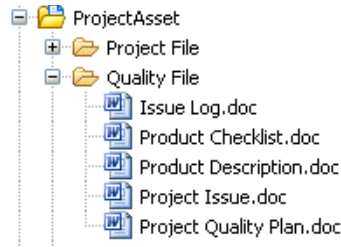
1. [Create a New Quality Process Project](#) to contain the templates.
2. Refer to [Using Quality Processes in Your Project](#) to associate a quality project with another project.

Task A Create a New Quality Process Project

1. Select **New > Project**, expand **Business Modeling**, select **Quality Process Project**, and click **Next**.
2. Enter a project name. Either accept the default location for the project (your workspace) or de-select the **Use default location** checkbox and click **Browse** to select a different location.
3. Click **Next**. In the resulting dialog, you can either create an empty quality archive project (into which you can copy your own template), or select Prince2 templates:



4. If you used the Prince2 templates, they are used to create the project:



If you created an empty project, copy the files and folders for your quality process.

Task B Using Quality Processes in Your Project

You can associate a quality process project with other project types in order to keep the documents expected by your quality team with other project assets. You can do this using one of the following methods.

From the New
Project Wizard

1. Create a project, selecting Business Assets as one of the asset types to be included in the project.
2. When prompted, specify the quality process project containing the correct templates (or accept the default).

After Project
Creation

1. After creating a project you can add business assets by selecting a standard folder and then **Special Folders > Other > Use as Business Assets**.
2. When prompted, specify the quality process project containing the correct templates (or accept the default).

Ad-hoc Assets

Ad-hoc assets include Microsoft Office documents and image files not recognized as project management assets. You can add business assets by right-clicking and selecting **New > Folder** or **New > File**, and selecting the appropriate **Parent Folder**.

To view the folder or files you have created, go to the Navigator view. In the Navigator view, you can also drag files and folders to different locations.

Working With Special Folders

Special Folders are folders in the Project Explorer that are reserved for storing specific types of assets. For example, the **Process Packages** folder is the default special folder for storing processes and the packages that contain them.

When you create a project, you have the option of creating a special folder for each type of asset that you include in your project. By doing this, you can utilize special features of the Project Explorer. For example, enabling a special folder for business processes allows you to view the participants, data fields and so on. If you do not use a special packages folder, you only see the XPDL file for the package in the Project Explorer. Another example of unique special folder behavior is the **Services** folder, which allows you to expand the operations of contained WSDL files:



The following table shows the asset types and the default special folder for each.

Default Special Folder	Asset Type
Business Assets	Prince2 templates, quality templates
Business Objects	Business object and concepts models
Forms	Forms generated by <i>TIBCO Business Studio Forms</i>
Organization	Organization models
Process Packages	Business process, packages
Business Objects	Business object model
Generated Services	WSDL file automatically generated for a business process
Service Descriptors	Imported WSDL files for services

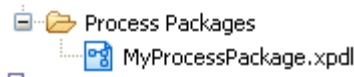
Enabling and Disabling Special Folders

You can disable a special folder by highlighting it in the Project Explorer, right-clicking, and selecting **Special Folders > Do not use as *SpecialFolderType***.

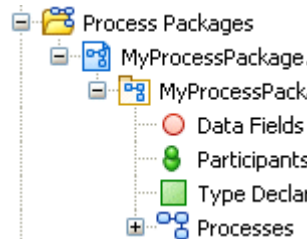


The following example uses the special folder for business processes as an example, however the special folders for other asset types are enabled and disabled in the same manner.

To disable the packages folder by highlighting the **Process Packages** folder in the Project Explorer and selecting **Special Folders > Do not use as Process Packages Folder**, which results in a view of the package like this:



Conversely, you can enable the use of the special packages folder by highlighting the packages folder in the Project Explorer and selecting **Special Folders > Use as Process Packages Folder**. This results in a view of the package like this:



Managing Project Lifecycle

When creating a project, you can assign it a version using standard Eclipse format:

`major.minor.micro.qualifier`

The specified version will be the default for project artifacts such as process packages and organization models, and can be used to indicate revisions to the project. The exact use of project versions should be coordinated with the solution designer working on the project, however the following are general guidelines:

major Incremented when the new version is incompatible with the previous version.

minor Indicates a compatible revision.

micro Indicates an internal change.

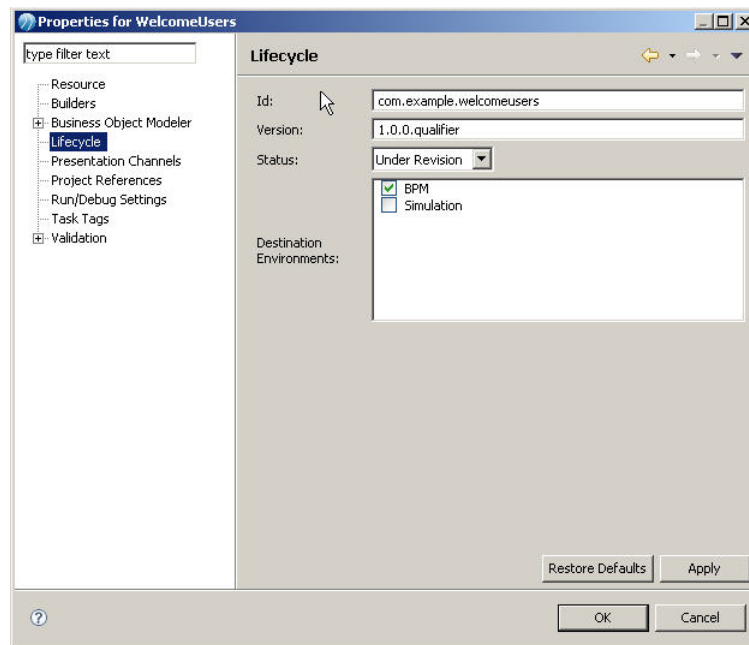
qualifier Used to identify unique builds may use time format or other convention.

Consult with your solution designer to establish a common practice.

Changing Project Lifecycle Settings

To change project lifecycle settings (such as a project version), do the following:

1. Right-click the project and select **Properties** and **Lifecycle**. The following dialog is displayed:

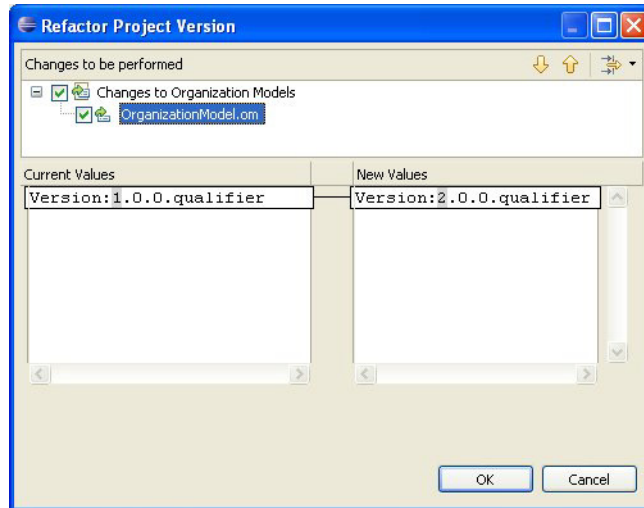


2. Enter the new settings and click **OK**.
3. Any new project artifacts will be created with the new settings. For example, if you changed the version, new process packages will be created with the new version. If there are project artifacts that were created with the previous settings that you want to update to the new settings, continue with [Refactoring Lifecycle Settings on page 61](#).

Refactoring Lifecycle Settings

The lifecycle settings of project artifacts (such as process packages or organization models) can be changed to match the settings of the parent project as follows:

1. Right-click the project and select **Refactor > Project Lifecycle...** A dialog is displayed that shows any project artifacts that can be refactored. When you select an artifact the settings that can be changed are displayed.



In this example, the project has the version **2.0.0.qualifier** and the organization model was created with the version **1.0.0.qualifier** (you can also refactor other settings such as status if they have changed).

2. To refactor the version of the organization model to match the version of the project, click **OK**.

Importing and Exporting Projects

You can import projects into TIBCO Business Studio, and export projects from it to archive files or file systems.

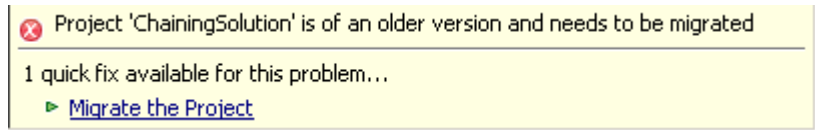


We recommend that you import or export individual projects, and do not attempt to import or export the workspace.

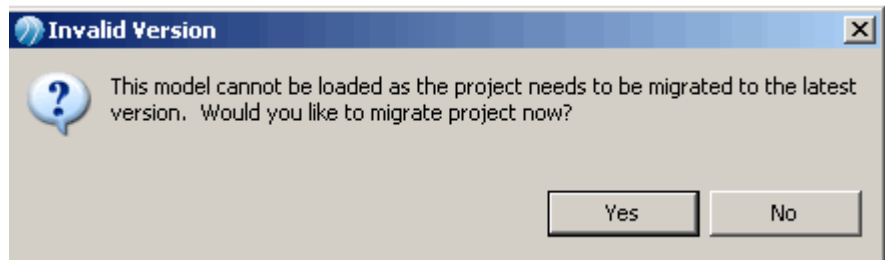


When you have imported projects into your workspace which were created in a previous version of TIBCO Business Studio, they may show a problem marker by the project name in Project Explorer. This occurs when a newer version of TIBCO Business Studio has different project configuration and/or file format requirements. The normal validation and build of projects will be disabled until this problem is fixed.

Select the project, and click **Ctrl +1**. You can then select one quick fix to automatically migrate the entire project (including all the BOM & XPDL files) to the current version, and clean (**Project > Clean**) and build it (**Project > Build Project**).



If you try to open an individual package within the project before you migrate the project, you will see a message similar to the following:



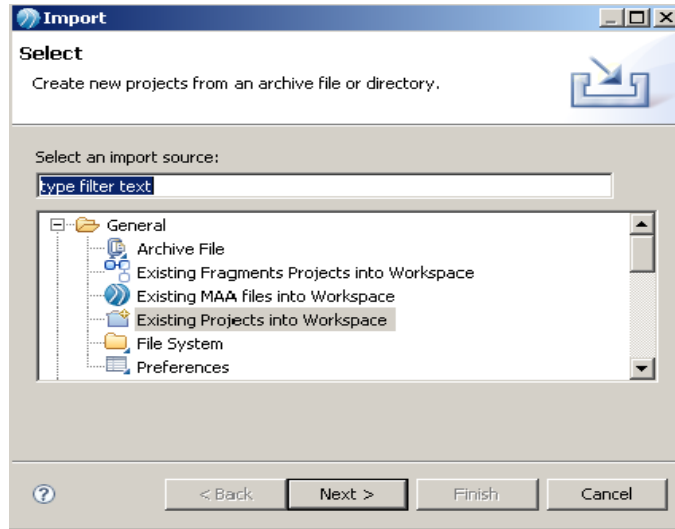
Select **Yes** and the quick fix will be applied to migrate the whole project.

Importing Existing Projects into Workspace

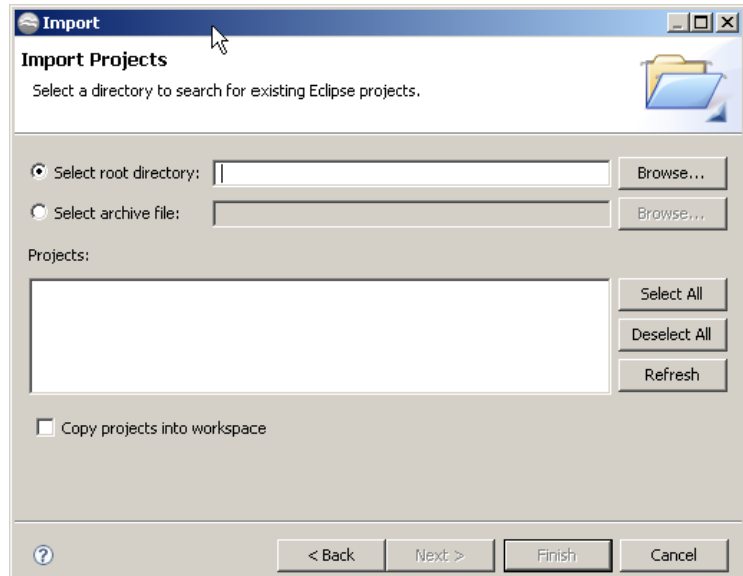
To import a project into the workspace:

1. From the File menu, select **Import**, and you see the **Select** page of the Import dialog.

2. Expand the General folder, select **Existing Projects into Workspace** and click **Next**.



3. Do one of the following:
 - Click on the **Browse** button for **Select root directory:** to locate other projects that are somewhere on the file system (for instance, in another workspace).
 - Check the radio button for **Select archive file:** and click on the **Browse** button to see projects that have already been archived which you may want to import.



4. Select the project you want to import, and click **Open**. The project is now selected.



This does not work if there is already a project of the same name in the workspace.

5. Select the **Copy projects into workspace** tick box and click **Finish**. The project is now imported and appears in Project Explorer.

Importing Existing MAA Files into Workspace

To import existing MAA files into the workspace:

1. From the File menu, select **Import**, and you see the **Select** page of the Import dialog.
2. Expand the General folder, select **Existing MAA Files into Workspace** and click **Next**.
3. Do one of the following:
 - Click on the **Browse** button for **Select root directory for MAA files:** to locate other MAA files that are somewhere on the file system (for instance,

in another workspace). This can be used to import multiple MAA files in one single go.

- Check the radio button for **Select MAA file:** and click on the **Browse** button to see MAA files that have already been archived which you may want to import. This is used to import one single MAA file at a time.
- 4. Select the MAA file you want to import, and click **Open**. The MAA file is now selected.



This does not work if there is already a project of the same name in the workspace.

- 5. Click **Finish**. The MAA file is now imported and appears in Project Explorer.

Importing Archive Files into Workspace

To import an archived project into the workspace:

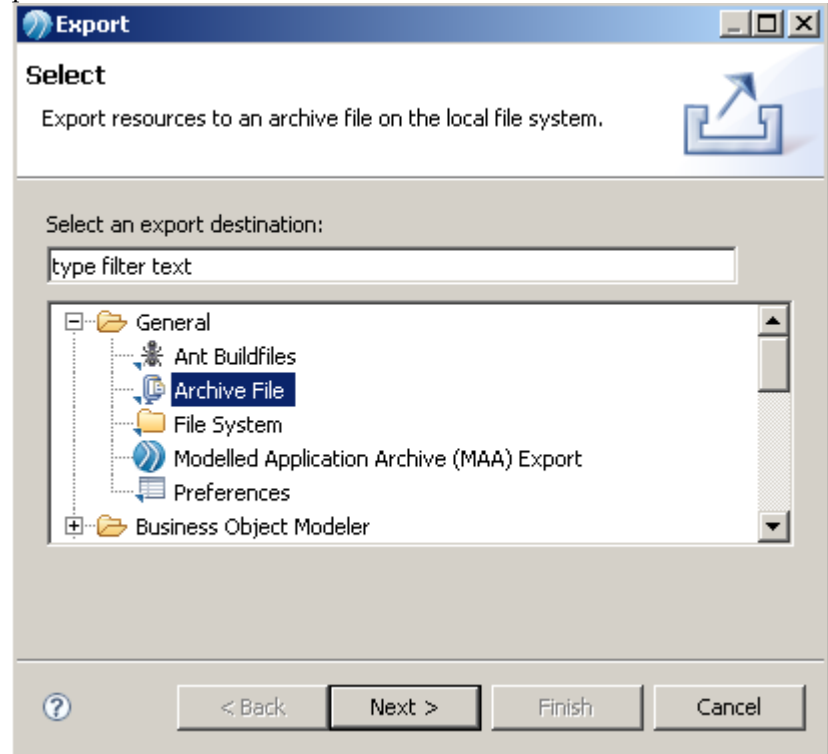
1. From the File menu, select **Import**, and you see the **Select** page of the Import dialog.
2. Expand the General folder, select **Archive File** and click **Next**.
3. In the **From archive file:** field, use the **Browse** button to locate the archive file you want to import.
4. In the **Into folder:** field, use the **Browse** button to identify the folder into which you want to import the file. This can be one of your existing projects. You can choose whether to check the **Overwrite existing projects without warning** tick box.
5. Click **Finish**. The archive file is now imported into the location you have identified.

Exporting Projects to Archive File

To export a project from Project Explorer to an archive file:

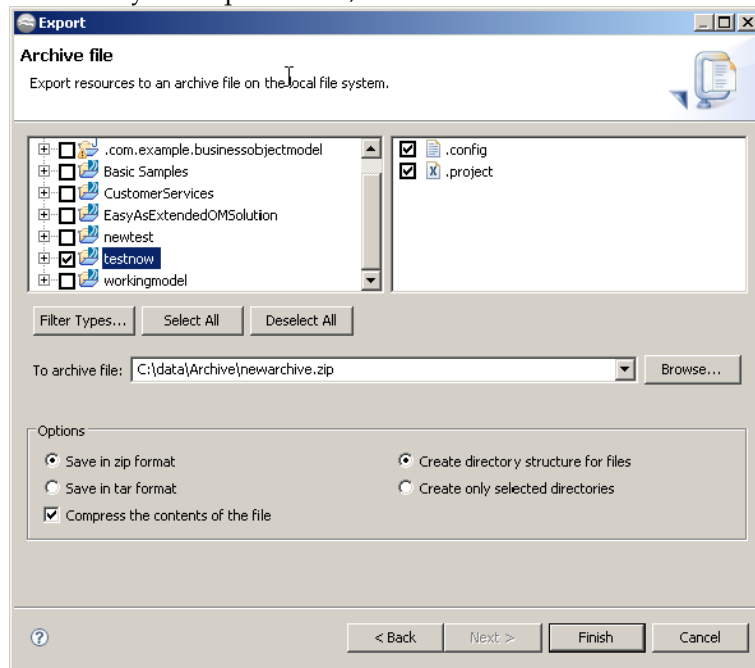
1. In your workspace, make sure you have the project you would like to export selected, and from the File menu, select **Export**. You see the **Select** page of the Export dialog:

2. Expand the General tab, select **Archive File** and click **Next**.



3. Select the project you want to archive from the list of projects available. Enter a location in the **To archive file:** box. You can use the Browse button to see existing archive files, and either use one of these or create a new one.

4. A number of archive options are available. Either accept the default settings or adjust them to your requirements, and click **Finish**.



Exporting Projects to File System

Perform the same steps as above - but at [step 2](#) select **File System**, and browse to the place you wish to save the files and click **Finish**.

Exporting Projects to Modelled Application Archive (MAA)

To export a project from Project Explorer to a Modelled Application Archive (MAA) file:

1. From the **File** menu, select **Export > Export as MAA file**. You see the **Export** page of the Modelled Application Archive (MAA)Export Wizard.
2. Select the project or projects (you can select multiple projects) you want to export to MAA, and the location you want to export them to. The default project location is **Exports/MAA** for each project.

Using Process Fragments

Rather than reusing an entire process, you can use the **Fragments** view to quickly create new processes. TIBCO Business Studio provides some predefined BPMN process fragments and you can also create your own process fragments. For example, there may be process patterns that you frequently use. By storing these patterns or "fragments" you can easily use them to construct new processes.

Migrating Fragments from Previous Versions

Any fragments that you created in previous versions of TIBCO Business Studio must be migrated to the current version as described in this section. The predefined fragments that TIBCO Business Studio provides do not need to be migrated because they are available in the current version.

To migrate fragments that you have defined, do the following:

1. Select **File > Import > General > Existing Fragments Projects into Workspace**, and click **Next**.
2. In the **Import** dialog, click **Browse** to select the directory where your fragments are located.
3. Select the fragments that you want to migrate, and click **Finish**.

Using the Predefined Fragments

TIBCO Business Studio provides predefined fragments in the folder **BPMN Process Fragments**. For example, **Basic Fragments** contains simple fragments such as an embedded sub-process, split conditional, and task sequence.

To use one of the predefined fragments, do the following:

1. Open the process.
2. Make sure that the Fragments view is visible. If it is not, select **Window > Show View > Fragments**.
3. In the Fragments view, there is a treeview that looks similar to the Project Explorer. This is the **Fragment Explorer**. Expand **BPMN Process Fragments** folder. This contains categories of BPMN process fragments.
4. Expand the category that contains the fragment you want to use.
5. Select the fragment.

6. Drag the fragment onto the process.

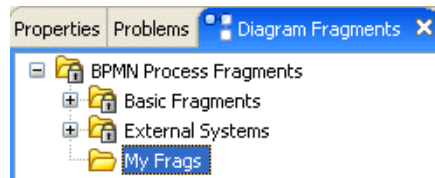


You can also use the **Copy** and **Paste** menu options either by right-clicking or by selecting from the **Edit** menu.

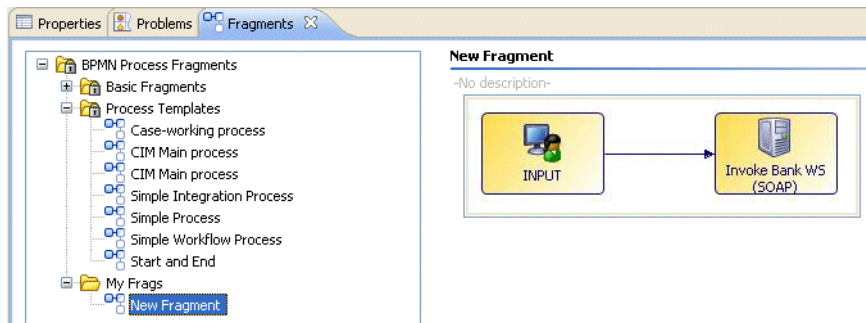
Creating Custom Fragments

If a portion of your process is one that you are likely to reuse in other processes, you can capture the useful part of the process and save it as a custom fragment as follows:

1. Create a new category by either right-clicking **BPMN Process Fragments** in the Fragment Explorer and selecting **New > New Category** or by clicking the **Create New Category** button (📁+).
2. Double-click in the name field of the newly added category and enter a name:



3. Do the following:
 - a. Holding down the Ctrl key, select the activities and sequence flows in the process that comprise the fragment.
 - b. Copy and paste the fragment into the Fragments view:



Using Custom Fragments

The Fragments that you create are stored in the category you specify and can be used to create new processes in the same way as pre-defined fragments (see [Using the Predefined Fragments on page 69](#)).



You can drag fragments and drop them into other categories.

Importing Custom Fragments

There are two ways to do this:

Import an Archive File

1. Create an archive file that contains the **.bsProjects** folder from your workspace. This folder contains your custom fragments.
2. The recipient of the archive file can then import it. To do this, select **File > Import > General > Existing Fragments Projects into Workspace**, and click **Next**.
3. In the **Import** dialog, select the **Select archive file** check box and click **Browse** to select the archive file that contains the fragments you want to import.
4. Click **Finish**. The fragments are imported into the current workspace.

Import the .bsProject Folder

1. Select **File > Import > General > Existing Fragments Projects into Workspace**, and click **Next**.
2. In the **Import** dialog, select the **Select root directory** check box and click **Browse** to select the location of the **.bsProject** folder from the file system.
3. Click **Finish**. The fragments are imported into the current workspace.

Using Source Control

One problem that occurs when dealing with processes across their normal life cycle (from creation, testing, rollout to maintenance), is how to know that a given process created by the analyst, elaborated by the solution engineer, and signed off by the process owner, is *exactly* the one that is in use in a specific environment (for example, a development, user-acceptance or production environment).

Particularly in large and complex projects where data is shared or modified by several people, a source control system becomes necessary. Most enterprises have one or more products for Source Configuration Management (SCM). This may be a commercial product such as Perforce, Rational's Clearcase or an open source solution such as:

- Concurrent Versions System (CVS) (<http://www.nongnu.org/cvs/>) or
- Subversion (<http://subclipse.tigris.org/>).

TIBCO does not provide its own SCM product, preferring to integrate with the enterprise's choice for SCM. The Eclipse feature for integrating with such an SCM product is known as the Team Synchronization. Since Eclipse provides CVS by default, the following section describes how to use Subversion; you should contact your SCM vendor for commercial plug-ins.



SCM is one part of Application Lifecycle Management (ALM) dealing only with the preservation of revisions of software at different times, not the editorial and approval processes that drive those different revisions.

Using Subversion with TIBCO Business Studio

This section describes how to use Subclipse for source control. Subclipse is an Eclipse plug-in that adds Subversion integration to the Eclipse environment, and is bundled with TIBCO Business Studio.



This section assumes that you or your Subversion administrator have installed an SVN server (see <http://subversion.tigris.org/>) or that you are using the version bundled with TIBCO Business Studio (Asset Central).

Right-click an existing project that contains items you wish to source control. Select **Team > Share Project**. Obtain the URL for the Repository Location from your Subversion administrator.

Creating a Project from an Existing Project in Source Control

1. Select **File > Import**. The Import wizard is displayed.
2. Expand **SVN** and select **Checkout projects from SVN** and click **Next**.
3. Set up the repository with the assistance of your Subversion administrator, (this should point to the parent directory to the one containing the project folder) and click **Next**.
4. Select the project folder and click **Next**.
5. Select the method for checking out the project and click **Next**.
6. Either use the default workspace or click **Browse** to check out the project elsewhere.
7. Click **Finish**. The project is checked out to your workspace.

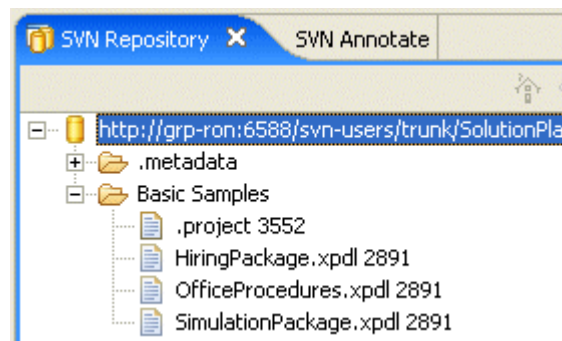


We recommend that you import or export individual projects, and do not attempt to import or export the workspace.

Browsing SVN Repositories

You can browse SVN repositories as follows:

1. Select **Window > Open Perspective > Other**.
2. Select **SVN Repository Exploring**.
3. This opens a perspective from which you can browse SVN repositories. For example:



Deleting Projects from SVN Repositories with Asset Central

If you are using Asset Central and want to delete a project that you have put in Subversion, it may appear that you have deleted the Project using Asset Central, however it remains in the Subversion repository until you delete it using another tool such as Tortoise SVN, the command line interface to Subversion, or a web-based tool.

Comparing Different Versions of a Process Package

You can compare the local Process Package you are working on with another version. This can be used to identify:

- any changes you have made since the last time you saved that Process Package. See [Two-way Compare](#).
- any changes made by another user in the Subversion copy since you last checked it out. See [Using the Process Package Comparison Editor with a Two-way Compare](#).



This ability to show a comparison graphically is also available for Task Libraries.

See [Using Source Control on page 72](#) for more information using source control to manage your processes.

Two-way Compare

To compare your current local copy of the Process Package with a previous local revision:

1. Select the Process Package .xpdl file in Project Explorer, and right-click it.
2. Select **Team > Show Local History**. This will show you a list of your revisions to the file.
3. Select the previous revision you are interested in comparing with your current version, and right-click **Compare Current with Local**.



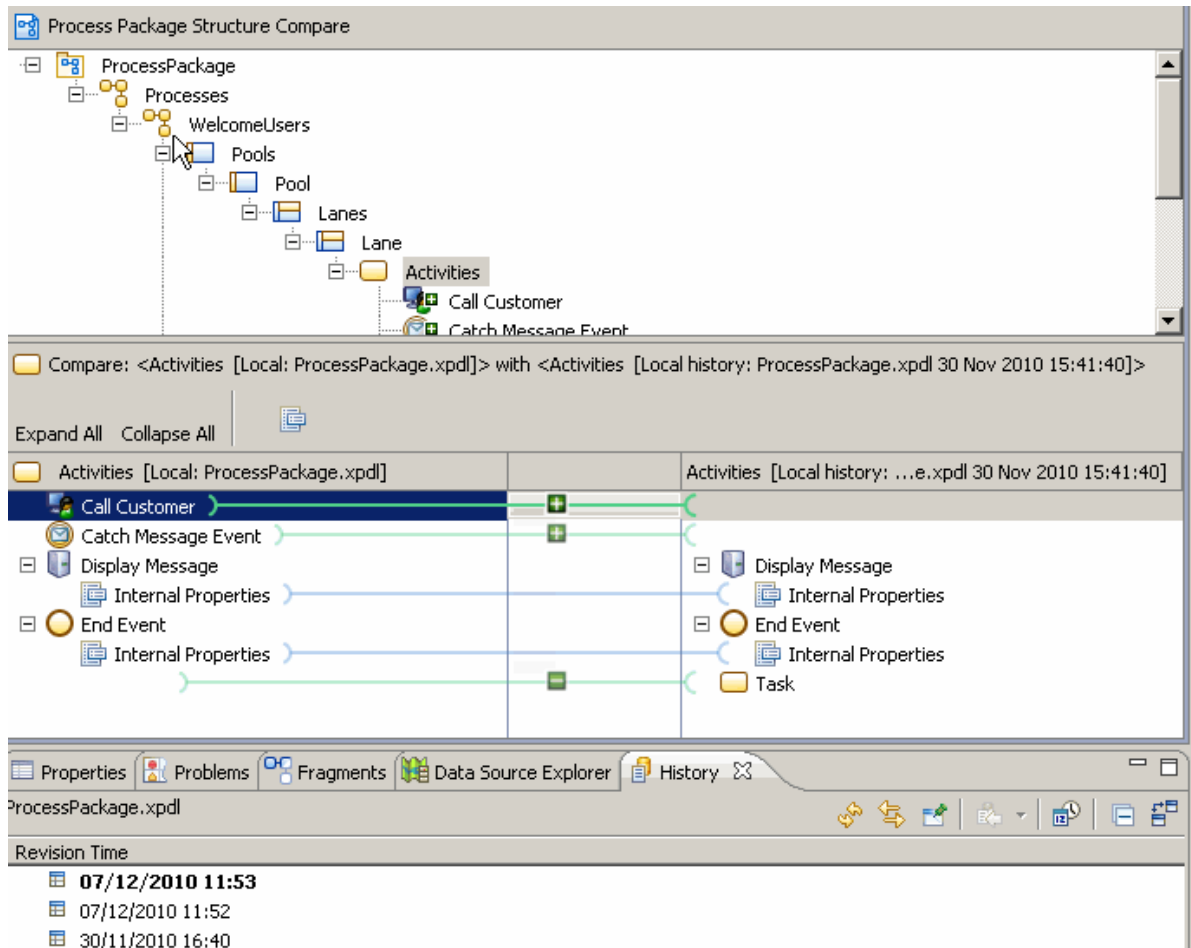
The default is to keep local history available for 7 days. If you need to see more history than this, select **Window > Preferences > General > Workspace > Local History** and edit the settings.

4. You can then compare the file you selected with your current local copy. The Process Package Comparison Editor is displayed showing you a tree of elements that are different in the two files. This is a **two-way compare**. In the Process Package Structure Compare window at the top of the screen, expand the Process Package and select the level you are interested in (for example, **Activities**).

The left/right revision views are populated with the tree structure of the different descendents of that element. The differences in your local copy will be shown in the left-hand column.

Using the Process Package Comparison Editor with a Two-way Compare

The following notes will help you interpret the results of a two-way compare. The screenshot below shows you an example of a two-way compare.



- The following icons indicate changes in a two-way compare:

As no common ancestor is available for comparison, all differences are treated as if changes were made on the left hand revision):



A green [+] (with no indication of direction) indicates elements that appear in the left hand revision that do not appear in the right hand revision.



A green [-] (with no indication of direction) indicates elements that appear in the right hand revision that do not appear in the left hand revision.

Items that do not have an icon by them are changed in either the left or right hand revision, or both (or have added, deleted or changed descendents).

- If you hover the cursor over the relevant icon, tooltip pop-ups will let you know that the information was added to the left or deleted to the right - so you may have added something, or the other version may have deleted something.

In a two-way compare is it not possible to identify which version the change was made in.

- In the Comparison view, you see Information elements (as indicated by the  icon. These elements are always shown regardless of whether there are any differences in the data that they represent. This is to give you a context for given elements and usually will contain the 2 or 3 major configurations for that element (such as task type, name and so on).
-  Internal Properties | Internal Properties (found at the level below a task, for example) is used to group a number of smaller changes made as the result of your main change/s under one heading. You are unlikely to need to examine these, but if you do want to see them in more detail select the  button above and to the right of the left/right content view.
- The central section of the comparison view shows connection lines between correlating objects on the left and right side. These include icons according to what the difference is (and on the side of the connection line where the change that caused the difference was made). The lines are colour coded to indicate the type of change that they represent (red=conflicting change, green=addition or deletion, blue = existing element changed).

Three-way Compare

A **three-way compare** is used to compare your changes to a Process Package with those of another user, as well as with the original (common ancestor) file on which both are based.

Typically, you might want to do this if another user checked a process package out of Subversion when you did, and they made changes and checked them back in. When you go to check your own version back in, Subversion identifies a potential conflict. To identify areas which the other user has changed and decide whether to adopt their changes and abandon yours, or overwrite them with your own, you can do a three-way compare.



Three-way compare is only available once a synchronize has been performed on the file (or its project). Synchronize is performed automatically for any file when it is committed to Subversion (to ensure that no conflicts exist with another user's changes).

See [Using Source Control on page 72](#) for information about using source control to control your project information.

To do a three-way compare (when you have checked a project file out of the source repository and made changes and saved them - and another user has done the same and committed their changes):

1. Attempt to commit your changes in the Team Synchronizing perspective.

You will discover that your changes conflict with those of another user and you will not be able to commit your changes until you have rectified this.

2. Select **Open in Compare Editor** to identify whether you want to adopt the changes from the other user, or commit your own changes instead.

Using the Process Package Comparison Editor with a Three-way Compare

The following notes will help you interpret the results of a three-way compare.

- The following icons indicate changes in a three-way compare:



A blue left-pointing arrow containing [+] indicates an element added in the right hand revision (as compared with the common ancestor).



A blue left-pointing arrow containing [-] indicates an element removed in the right hand revision (as compared with the common ancestor).



A plain blue left-pointing arrow indicates an element changed in the right hand revision (as compared with the common ancestor).



A grey right-pointing arrow containing [+] indicates an element added in the left hand revision (as compared with the common ancestor).



A grey right-pointing arrow containing [-] indicates an element removed in the left hand revision (as compared with the common ancestor).



A plain grey right-pointing arrow indicates an element changed in the left hand revision (as compared with the common ancestor).



Red arrows pointing both ways indicate a conflict: the element or its descendants were changed in both the left hand revision and the right hand revision.

- If you hover the cursor over the relevant icon, tooltip pop-ups will let you know that the information was added to the left or deleted to the right - so you may have added something, or the other version may have deleted something.

In a three-way compare, you can distinguish whether an additional element was caused by an addition by someone else or a deletion by you. This is because the comparison can check whether the additional element existed in the common ancestor, and if it did not, then you will know whether you have changed it or whether it was another user.

- In the Comparison view, you see Information elements (as indicated by the icon. These elements are always shown regardless of whether there are any differences in the data that they represent. This is to give you a context for given elements and usually will contain the 2 or 3 major configurations for that element (such as task type, name and so on).
- Internal Properties | Internal Properties (found at the level below a task, for example) is used to group a number of smaller changes made as the result of your main change/s under one heading. You are unlikely to need to examine these, but if you do want to see them in more detail select the button above and to the right of the left/right content view.
- The central section of the comparison view shows connection lines between correlating objects on the left and right side. These include icons according to what the difference is (and on the side of the connection line where the change that caused the difference was made). The lines are colour coded to indicate the type of change that they represent (red=conflicting change, green=addition or deletion, blue = existing element changed).

Emailing Processes

The XPDL package files and the processes contained within can be sent using email.

To send a Process:

1. In the Project Explorer, find the name of the package that contains the process you want to send.
2. Attach the package file to the email message using a method such as copy and paste or drag and drop.

If you receive a Process:

Get the Process into TIBCO Business Studio as follows:



Be careful of overwriting existing packages with the same name.

- Save the attachment directly into your workspace and refresh the Project Explorer.



The default workspace location is **C:\Documents and Settings\username\workspace**. However, you may have selected an alternate location when you first started TIBCO Business Studio.

- Save the attachment to a file then do one of the following:
 - Drag the package file from Windows Explorer into a packages folder in the Project Explorer in TIBCO Business Studio.
 - In Windows Explorer, press **Ctrl+C** to copy it, return to TIBCO Business Studio, and press **Ctrl+V** to paste it into the Project Explorer, into a process packages folder.

Creating Documentation

From within TIBCO Business Studio you can export information about your entire project, or your Organization Model, Business Object Model, or Process and the Package that contains it, to an HTML file.

You can also use the command line to create documentation.

To Create Documentation from within TIBCO Business Studio

1. You can do one of the following:
 - Right click on a model resource (for example: an **.xpdl**, **.bom** or **.om** file).
 - Right click on a project.
 - Right click on a 'special' folder.
2. You can do one of the following:
 - If you selected a model resource or the whole project, select **Export > Documentation**.
 - If you selected a 'special' folder, select **Export > Export > Documentation**.
3. One of the following will happen:
 - If you selected a model resource in [step 1](#), it is selected in the wizard (you can change the selection here if required).
 - If you selected the entire project or a 'special' folder in [step 1](#) all possible resources are selected in the wizard.)
4. You have the option to select the destination for the documentation files.
 - The default location for generation is `<ProjectPath> > Documentation > * Model Documentation`.
 - The default for the Index page is `<ProjectPath> > Documentation`.



You can specify a location outside the workspace. In this case, the location must exist or an error will appear in the wizard)

5. Click **Finish** to generate documentation for the selected model file(s) and generate an index page containing overview information for the selected files(s).

To Create Documentation from the Command Line

1. Close the workspace that contains the project for which you want to generate documentation.
2. Navigate to the directory that contains the TIBCO Business Studio executable file. For example:

```
C:\Program Files\TIBCO\BusinessStudio
```

3. Open a command window.
4. Enter a command line including specification of:
 - Workspace
 - Project name
 - Output location

This will generate documentation for the selected file(s) and also an index page.

An example of command line invocation would be:

```
%tibco home%\studio-doc.exe -data <workspace> -projectName  
<project name> -outputPath <output path>
```

To Generate Documentation On Demand

You can generate documentation on demand from your local filesystem, without requiring a workspace or TIBCO Business Studio project.

To do this, you require a supported file type (.maa) on the local filesystem. An .maa file is a way of compressing a TIBCO Business Studio project.

To create an .maa file, in the Project Explorer in TIBCO Business Studio, go to your project, and select the Documentation folder. Then right click to select **Export > Archive File**. When naming the file, give it the suffix **.maa**.

To generate documentation on demand do one of the following when opening an .maa file:

- If the file is already associated with the Documentation tool, double-click the desired file within file system explorer.
- If the file is not already associated with the Documentation tool, right click on the file and choose "Open With.." and point to the Documentation tool (<Business Studio Home>/studio-doc.exe).

The command line generates documentation in the same path as the .maa file for the selected file(s) and also an index page.

If an output path is provided in the *<Business Studio Home>/studio-doc.ini* the documentation is generated in the path provided. To add an output path, add the following to the beginning of the *studio-doc.ini* file:

```
-outputPath  
<OutputPath>
```

To View Documentation

1. Export the documentation as described in [To Create Documentation from within TIBCO Business Studio](#).
2. If you created the documentation from within TIBCO Business Studio:
 - In the Project Explorer, expand **Documentation**.
 - Right-click the HTML file for the package and select **Open With > Web Browser**.

If you created the documentation from the command line:

- Navigate to the directory where you created the HTML file and open it in a web browser.
 - If you created the HTML file in a directory in your workspace, you can refresh the Project Explorer, right-click the HTML file for the package and select **Open With > Web Browser**.
3. Note that the sections of the documentation are hyper linked as well as the image of the process (for example, clicking a task displays the section that describes that task).

See "Deploying BPM Applications" in the *TIBCO Business Studio BPM Implementation* guide for more information.

Using Presentation Channels

Presentation channels control how tasks will be displayed to the user after deployment.

When you use Presentation Channels, you can choose between Workspace Google Web Toolkit, Workspace Email and Openspace Google Web Toolkit which are provided by default. You can also choose to add channel types, or create your own presentation channel. See the *TIBCO Business Studio BPM Implementation Guide* for more information.



This is ignored for those with iProcess as their destination environment.

Editing Email Attributes at Workspace Level

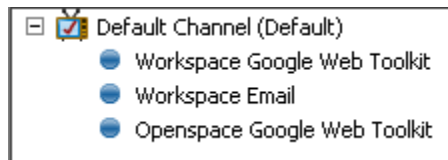
You can edit the attributes of the Workspace Email channel type at the workspace level or the project level.



Project level configuration overrides workspace channel configuration.

To edit at the workspace level:

1. Select **Window > Preferences > Presentation Channels**.
2. Expand **Default Channel (Default)** to view the presentation channels available.



3. Click on **Workspace Email** to view its attributes. The attributes are as follows:

Attributes	Extended Attributes	Properties
Name	Value	
✦ mailTemplateLocation		
✦ mailSubject		
✦ mailPriority	3	
✦ fromAddress		
✦ ccAddress		
✦ mailHeaders		
✦ attachments		
✦ presentationChannelId	GIGWTPull_DefaultChannel	

You can edit the values of the Email attributes to meet your own requirements:

mailTemplateLocation	This attribute allows to specify a custom email template. It is currently TIBCO branded. It contains a picker, () to allow you to tailor it to your own requirements. See Creating an Alternative Email Template on page 87 .
mailSubject	A brief summary of the contents of the message.
mailPriority	An entry is required here. The default setting is 3. Settings can range from 5-1, with 1 being the highest priority and 3 being what is considered Normal.
fromAddress	Address where the message originated from, including an optional name, for example, Jane Smith jsmith@anycompany.com. This is a mandatory field. This needs to be a valid address, depending on your SMTP server settings. Refer to your SMTP Server documentation for more information.

ccAddress	An additional delimited list of recipients. This is an optional field.
mailHeaders	List of custom headers, of the format <i>Header Name:Header Value</i> . This is an optional field.
attachments	Reserved for future use.
presentationChannelId	This is the channelId used when the user clicks the HTML link in the pushed email message. The default setting is GIGWTPull_DefaultChannel. This is a mandatory field.

The following "tokens" can be put into any of the above Email attributes:

- **%%token.workItemId%%** - The work item id.
- **%%token.entityName%%** - The entity name who requires push notifications as defined in the Organizational Model.
- **%%token.mailDate%%** - The date and time the pushed mail message was sent.

You can add a token by typing the string directly into an attribute value. For example, for **mailSubject** you could enter:

PUSH DEMO - Pushing Work Item Id %%token.workItemId%%

Then at runtime, the token will be replaced by the actual value:

PUSH DEMO - Pushing Work Item Id 1

Editing at Project Level

1. From Project Explorer, right-click on the project and select **Properties > Presentation Channels**. Check the tickbox for **Enable project specific settings**.
 - or select **Window > Preferences > Presentation Channels** and click on **Configure Project Specific Settings...** Select the project you want to configure and press **OK**.
2. Expand **Default Channel (Default)** to view the presentation channels available

3. Select **Workspace Email** and edit the attributes as described above.
4. Click **Apply**. The attribute changes are applied next time you deploy your project.

To restore default values you can click **Restore Defaults**.

Extended attributes are also available for custom use.

Creating an Alternative Email Template

You can create your own email template which can contain organization-specific information, with your own logos and corporate style. After deployment it can be used as the default email template for any Push Destination using the **Workspace Email** channel type.

The files you can use to create your email template are as follows:

- **HTML file** (Mandatory) This contains the information about what you want the email to contain. It typically includes:
 - a reference to the .css file as its stylesheet. This can be included using an html line similar to the following:


```
<link rel="stylesheet"
type="www.tibco.com/stylesheet/test.css"
href="www.tibco.com/stylesheet/test.css" />
```
 - references to any of the graphics files contained in the cid.properties file (referred to by their cid: identifier, prefaced by 'cid:') , for example:


```

```

refers to `NewProductLogo.png` in the example in cid.properties above. Referring to them in this way means they can be shown externally as part of the email when the html file is used to create an email after deployment.
 - the content you want to include in the email.

This html can contain any of the following tokens, each of which will be replaced by an actual value at runtime;

"%%token.workItemUrl%%"	URL of the work item.
"%%token.workTypeDescription%%"	Description of the work type.
"%%token.workItemId%%"	The id of the work item.
"%%token.entityName%%"	The entity name who requires push notifications as defined in the Organizational Model.

%%token.mailDate%%"	The date and time the pushed mail message was sent.
%%token.mailFrom%%"	The name of the sender of the pushed mail message.
%%token.workItemAllocationMethod%%"	The allocation method used for the work item. For example, Allocate Work Item(s) To World.
%%token.workItemStartDate%%"	The start date for the work item.
%%token.workItemEndDate%%"	The end date for the work item.
%%token.workItemPriority%%"	The priority of the work item.
%%token.workItemScheduleStatus%%"	The schedule status of the work item. For example, DURING.
%%token.mailSubject%%"	The subject line for the pushed mail message.
%%token.mailTo%%"	The user/s who the pushed mail message is sent to.
%%token.mailCc%%"	The user/s who the pushed mail message is copied to.
%%token.mailBcc%%"	The user/s who the pushed mail message is blind-copied to.
%%token.workItemState%%"	The state of the work item. For example, ALLOCATED.
%%token.processActivityId%%"	The id of the process activity.
%%token.processTemplate%%"	The name of the process template.
%%token.processApplicationId%%"	The id of the process application.
%%token.workItemGroupId%%"	The Group id for the work item.
%%token.hostIPAddress%%";	The IP address of the host.
%%token.hostMachineName%%"	The host machine name.
%%token.baseurl%%"	The base URL.

- **CSS (Cascading Style Sheet) file** This is a stylesheet file, typically for your organization, which contains standard information such as banners or background colors, that you want to be in every email you send.
- **graphics** you are referencing from the `cid.properties` file, which would then be used by the html file. These must be available locally, and can either be in the same directory (**Presentation Resources**), or in a sub-directory.



- **cid.properties** This allows you to reference any graphics files you have available locally (in the same directory, or a subdirectory) which you want to be used when you push the work item (which then sends an email to notify the user of the work item). It is specifically used for embedded images / css in emails, rather than providing the option of external calls.
- Each filename will have a **cid:<name>** identifier which allows it to be accessed remotely as well as locally. For example, the `cid.properties` file might contain lines such as the following, where `cidOne` is the `cid:` identifier for `NewProductLogo.png` when you reference it later in the HTML file:

```
cidOne:NewProductLogo.png
```

To create your Email template files:

In Project Explorer, navigate to the project you want to add the Email template to:

1. Right-click **New > Folder** and name your folder.
2. Right-click on the folder you created, and select **Special Folders > Use as Presentation Resources Folder**.
3. From the Presentation Resources folder you created, right-click **New > File** and create the files you require.

Alternatively:

- Right-click **Presentation Resources > New > File**, and create the files you require



If you create the files you require elsewhere, copy and paste them by copying them and then clicking on **Presentation Resources** (or your new Presentation Resources folder) > **Paste**.



This email template will be available to this project only. If you attempt to access it from another project, you will receive a validation error, as the files contained are not available outside this project.

To apply the Email template to a Project:

1. From the project, right-click **Properties > Presentation Channels**. Select the checkbox **Enable project specific settings**.
 - or select **Window > Preferences > Presentation Channels**, and click **Configure Project Specific Settings...** Select the project you want to configure and click **OK**.
2. Expand **Default Channel** and select **Workspace Email**.
3. Click in the **Value** field for **mailTemplateLocation**, and you will see the picker, ():
4. Use the picker, () to navigate to the HTML file you created in the Presentation Resources folder, and click **OK**.
5. Click **Apply**.

Next time you push an email, it will use this template.



Email templates are scoped to one project only. If the template is applied on the workspace level, then all relevant projects should have the specified custom template files available inside them. It is then recommended to overwrite the workspace Presentation Channel definition on the project level when the custom email template needs to be provided.

Chapter 5

Working with Activities

This section describes some of the common tasks that you perform using TIBCO Business Studio Activities.

Topics

- [Activity and Task Overview, page 92](#)
- [Request Response Operation with Send and Receive Tasks, page 97](#)
- [Creating Scripts, page 98](#)
- [Sub-Processes, page 100](#)
- [Activity Markers, page 102](#)
- [Resource Patterns and Work Distribution, page 108](#)
- [Making Processes Easier to Follow, page 118](#)
- [Swimlanes, page 121](#)

Activity and Task Overview

An **activity** represents work that a company or organization performs using business processes. An activity can be atomic (it is not broken down into a finer level of detail) or non-atomic. Atomic activities are represented in the Process Editor by tasks. For more information about how BPMN defines activities and tasks, see <http://www.bpmn.org>.

When creating the TIBCO Business Studio Process, each time a different person, group, role, or system does something, an activity is added to the Process.

Activities may be triggered by events such as the receipt of an email, phone call or workflow item, and may involve making a judgement on the presented facts and performing an action (such as entering data to a computer system, phoning someone in the same or a different organization, and so on).

A task in a process diagram represents an atomic activity (one that cannot be further broken down). A task of an unspecified type looks like this in the Process Editor:



If the activities can be broken down into finer steps, they should be represented as **Sub-Processes**. The Activity Type is set in the **Properties** view.

User Tasks

User tasks are those that require human interaction with a software application. For example:



User tasks can be further configured for inbound and outbound parameters. Forms can be generated from the task's input and output parameters, representing the information you want to present to and capture from the user.

You can also generate a pageflow process from a task [Creating a Pageflow Process From a User Task on page 104](#).

TIBCO Business Studio Forms enables you to design, view, and test the forms you need to collect user input in a business process. You can create sophisticated forms without programming, and associate them with user tasks in order to provide richer user experiences for business process participants.

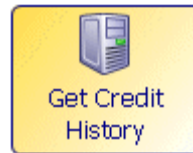
Manual Tasks

Manual tasks are those that are completed by a person without using software. For example:



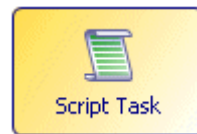
Service Tasks

Service tasks can ideally complete without human interaction (for example, an automatic email notification or a web service). For example:



Script Tasks

Script tasks contain a set of instructions written in a scripting language (usually added to the step by the solution engineer) that will be executed in the runtime environment when the process is deployed and executed. For example:



Using the business analysis capability, you can add text to a script task to describe the desired behavior of the script. However, by switching to solution design capability or by clicking **Provide Implementation Details** in the Properties view for the task, you can enter JavaScript to a script task. For more information, see the relevant implementation guide.



Script tasks and service tasks can be used to achieve the same results, however TIBCO recommends that only small and simple tasks are implemented using script tasks because properly-written services are easier to maintain and test.

Send Tasks

Send tasks are used to send messages to a system or person outside of the process (often using a web service):



They can be paired with a receive task or message event to form a request response operation (see [Request Response Operation with Send and Receive Tasks on page 97](#)).

Receive Tasks

Receive tasks are used to wait for a message from a system or person outside of the process (often using a web service). This type of task can be used to start a process as long as it has no incoming sequence flow and there are no start events in the process:



On the **Interface** tab, you can add parameters to a receive task; however, you cannot add data fields because data fields are used internally in a process, and parameters are in this case, inputs from an external process (for more information, see [Creating Data Fields and Parameters on page 164](#)).

Receive tasks can be paired with a send task or message event to form a request response operation (see [Request Response Operation with Send and Receive Tasks on page 97](#)). You must also ensure that incoming messages are received by the correct process instance. For more information, see [Handling Incoming Messages Using Correlation Data on page 169](#).

Reference Tasks

Reference tasks refer to another task and prevent you from having to duplicate the same task several times in a process.

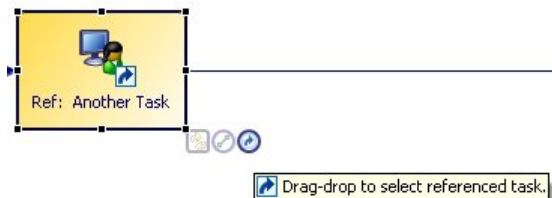


You can create a reference to another task using either the **Reference Task** gadget on the reference task, or in the Properties view for the reference task.

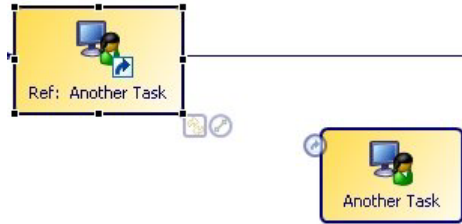
Creating a Reference with the Reference Task Gadget

A gadget is a graphical tool that allow you to easily create references. You can create references to tasks on the diagram using a gadget as follows:

1. Place a reference task on the diagram.
2. Select the reference task.

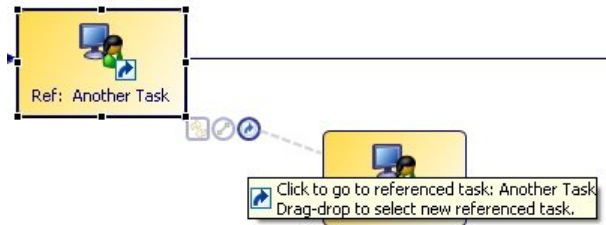


3. Drag the **Reference Task** gadget to the task to which you want to refer.



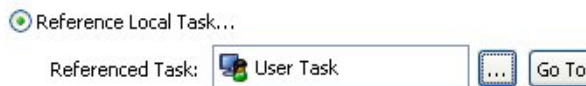
If the task to which you want to refer is not in the visible area of the Process Editor, drag the gadget to the right-hand edge of the Process Editor. After a short pause, the Process Editor scrolls to the right. Alternatively, select **Diagram > Zoom Out** until you can see the task to which you want to refer.

4. Once you create a reference, you can click the **Reference Task** gadget to go to the referenced task, or drag the **Reference Task** gadget to create more references:



Creating a Reference in the Properties View

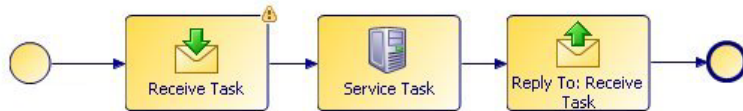
To create a reference to another task on the **General** tab of the Properties view for the task, complete the **Referenced Task** field by clicking  and selecting a task from the list:



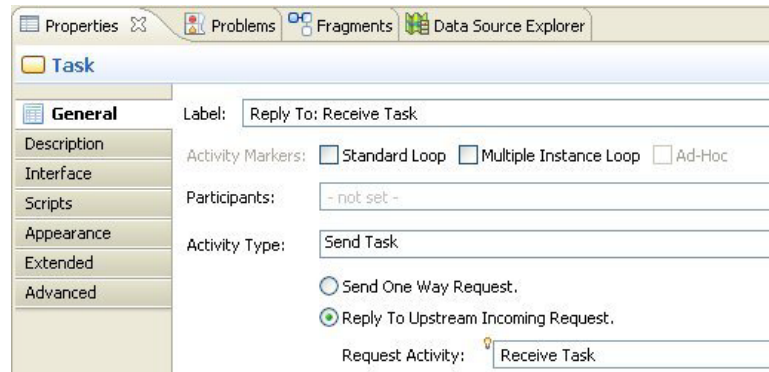
After you have created a task reference, clicking **Go To** selects the target of the reference task and brings it into the visible part of the Process Editor (it also displays the Properties view for the task). To return to the previously-selected task, click  on the toolbar.

Request Response Operation with Send and Receive Tasks

To configure a process to receive a request-response message from an external application, and then to reply to the request, use message events, send and receive tasks, or a combination thereof. For example,



In this example, because the send task is configured as a reply to the receive task in the Properties view:



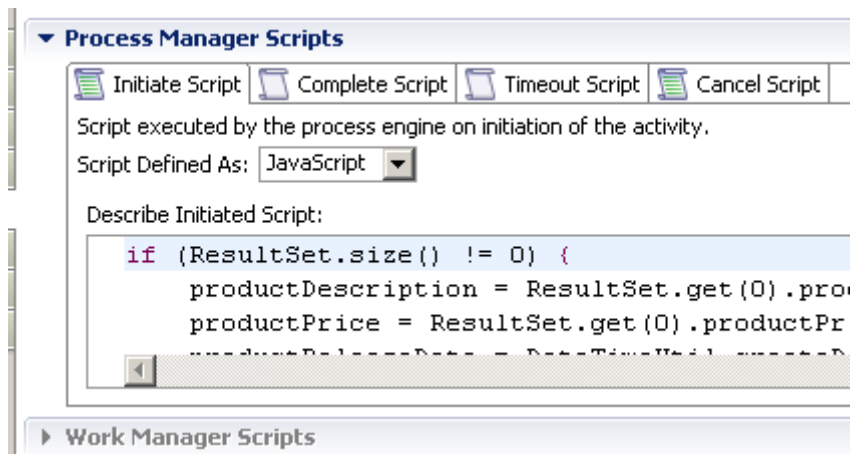
For more information about using message events, see [Implementing Message Events on page 158](#).

Creating Scripts

All types of task can have scripts that write information to a destination-specific system. User tasks can have scripts that are run for example, when the work item is opened or closed.

The **Scripts** tab allows you to add text that describes these scripts. It is the task of the solution engineer to translate the outline of the script into a scripting language supported by the selected destination environment. To add scripts, do the following:

1. Click the task to which you want to add a script.
2. In the Properties view for the task, click the **Scripts** tab.
3. The Scripts available are divided into two sets, **Process Manager Scripts** and **Work Manager Scripts**. Open the set you require (if one set is not available to you for this task, it will be grayed out)

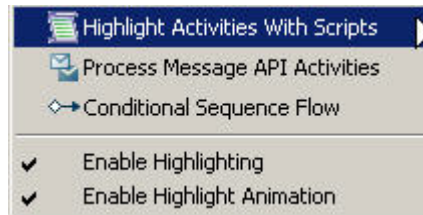


4. Click the tab for the desired type of script. Any tabs that have scripts defined have a script icon before the script name. Tabs with no script defined have an empty script icon. So in the example above, a script is defined for the **Initiate Script** and **Cancel Script** tab.
5. Select **Free Text** from the **Script Defined As** list. Describe your script in the area provided. For more information about how the various types of scripts are implemented, see the appropriate implementation guide.



You can highlight tasks in a process that contain scripts. Click in the Process Editor for the process. On the toolbar, you will see a button: 

Click on this to see the dropdown, and select **Highlight Activities with Scripts**.



This will highlight all activities in a process that contain scripts, and all other activities will be grayed out.

Sub-Processes

Some activities can contain further steps, or sub-processes.

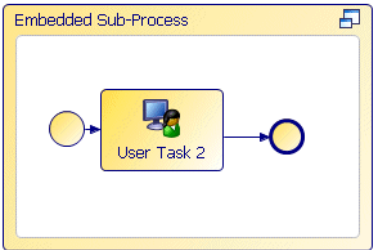
There are two types of sub-process that are described in this section, embedded and reusable. For more information about how processes can be reused, see [Using Sub-Processes and Process Interfaces on page 189](#).



A sub-process that is embedded can be re-factored into a reusable sub-process and vice versa. For more information see [Refactoring Sub-Processes on page 197](#).

Embedded

An embedded sub-process looks similar to the following:



An embedded sub-process has the following characteristics:

- It is fully contained within the parent process, and is executed within the parent process.
- Activities within the embedded sub-process have access to the same data fields and parameters as the parent process and package.
- No data mapping is required.
- It cannot contain lanes and pools.

Reusable

Activities that contain a reusable sub-process look like this in the Process Editor:



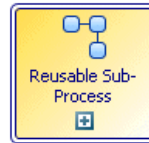
Click the plus sign (+) in the activity to view the sub-process.

An activity of this type defines a call-out to another process:

- The called process exists as a separate process from the parent process, and because of this it can be started from other processes.
- The called process does not have access to data fields and parameters of the calling process and package. for this reason, data mapping must be performed to and from the called process.

Transactions

A sub-process (either embedded or independent) can be specified as a transaction using the **Is a transaction** check box. This means that the behavior of the sub-process is governed by a transaction protocol (defined in the runtime environment). This is indicated by a double-line around the activity:



Typical outcomes of a transaction sub-process that you should cater for are success and cancel. For more information, refer to the BPMN specification (see <http://www.bpmn.org>).

Activity Markers

You can select BPMN Activity Markers on the Properties of the activity. The currently selected Activity Marker is indicated by the following symbols on the activity:

- **Multiple Instance Loop**  Indicates a task or sub-process that is replicated a fixed number of times based on the evaluation of an expression. The ordering can be either parallel or sequential (see [Creating a Multi-Instance Loop on page 129](#)).
- **Standard Loop**  Indicates a task or sub-process that may have more than one instance, depending on the conditions of the loop. A standard loop consists of a Boolean expression that is evaluated before or after each cycle of the loop. If the expression evaluates to True, the loop continues.

The conditions of the loop are set on the **Loops** tab (for more information, see [Using Loops on page 128](#)).

- **Ad-hoc**  Indicates an embedded sub-process marker that contains activities that have no pre-defined sequence. This also means that the number of times the activities are repeated is completely determined by the performers of the activities and cannot be defined beforehand.

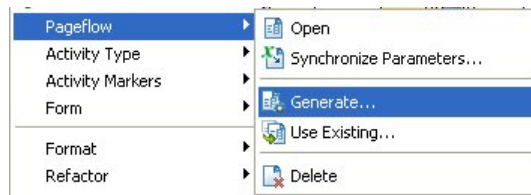
Working with User Tasks and Pageflow Processes

In addition to creating a standalone pageflow process, you can do the following from a user task:

- Select an existing pageflow process.
- Create a pageflow process that is referenced from the selected user task.

In the runtime environment, the pageflow process is run when the work item associated with the user task is opened.

When you right-click a user task, the **Pageflow** menu provides the following options:



- **Open** Select this option to open the associated pageflow process in the Process Editor (if a pageflow process has already been selected for the task).
- **Synchronize Parameters** Select this option if you have made changes to the parameters in the user task, and want the changes reflected in the corresponding parameters in the associated pageflow process (see [Synchronizing Parameters With a Pageflow Process on page 106](#)).
- **Generate** Select this option to generate a new pageflow process that will be associated with this user task. The parameters associated with the user task are replicated for the pageflow process.
- **Use Existing** Opens the **Select Form** dialog from which you can select an existing pageflow process.
- **Delete** Deletes the reference from the user task to the pageflow process. The pageflow process itself is not affected.

Selecting a Pageflow Process

If you have already created a pageflow process, you can select it from a user task as follows:

1. Select the user task.

2. Do one of the following:

- On the **General** tab for the user task, select the Pageflow option and click  to display the **Select Pageflow Process** dialog.



Expand the projects to select a pageflow process and click **OK**.

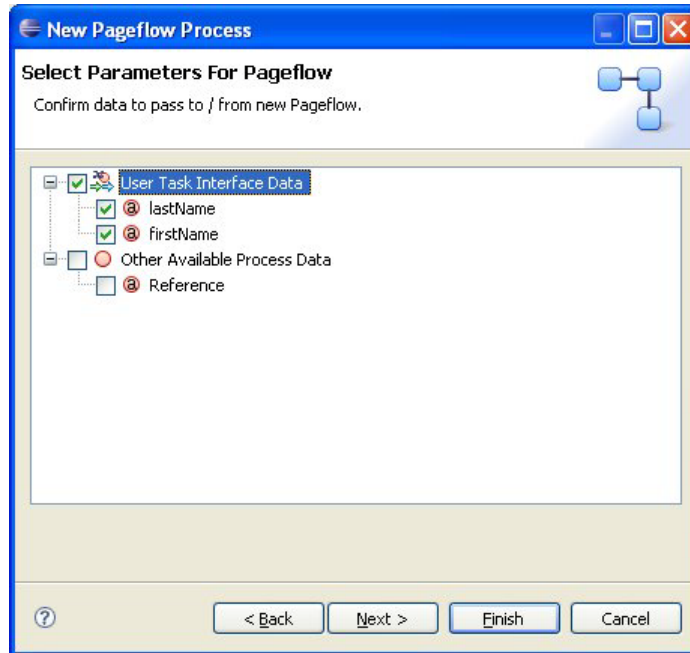
- Right-click the user task and select **Pageflow > Use Existing** to display the **Select Pageflow Process** dialog. Expand the projects to select a pageflow process and click **OK**.
3. The **Pageflow** field is automatically completed with a URL that points to the pageflow process (relative to the **Forms** special folder). You can open the pageflow process by clicking **Open Pageflow Process**.

Creating a Pageflow Process From a User Task

To create a pageflow process that is associated with a user task, do the following:

1. Select the user task to which you want to associate a pageflow.

2. Right-click and select **Pageflow > Generate**. The **New Pageflow Process** dialog is displayed.



3. Confirm the data that you want to associate with the pageflow process and click **Next**:
 - **User Task Interface Data** Select from the process data specified on the **Interface** tab of the user task. If no explicit process data is selected on the **Interface** tab, all process data is available.
 - **Other Available Process Data** Select from the process data that is not associated with the user task.



The parameters that you associate with the pageflow process are available to user task forms after the pageflow process is created.

4. Enter the **Label** of the process. If you want to use a template to create the process, select the template and click **Finish** to create the pageflow process or **Next** to specify additional options. Subsequent dialogs are the same as those described in [Creating a Pageflow Process on page 46](#).

Synchronizing Parameters With a Pageflow Process

When a pageflow process is first generated from a user task, a dialog is displayed that allows you to create the pageflow process with its own set of parameters (either replicating the existing data fields and parameters of the user task, or a subset thereof). User tasks in the pageflow process can display forms that utilize these parameters.

If you add or remove new parameters to the user task in the parent process (after the pageflow process is generated), you can update the list of user task parameters that are known to the pageflow process as follows:

1. Right click the user task, and select **Pageflow > Synchronize Parameters**. The **Synchronize Pageflow Parameters With User Task** dialog is displayed:



2. Confirm the changes and click **Finish**. For example, if parameters have been added to the pageflow process but not the user task, synchronizing deletes them from the pageflow process. If parameters have been added to or removed from the user task, synchronizing adds or removes the corresponding parameters in the pageflow.

After synchronizing, the parameters in the Project Explorer listed for the pageflow process should be identical to the parameters listed for the parent process.

Mandatory Parameters and Pageflow Processes

When a pageflow process has a mandatory parameter, that parameter must have a value before the pageflow can be considered complete. This is different to sub-process parameters, where the mandatory flag controls whether mapping to the parameter is required or optional.

This means that a pageflow process and associated user task are expected to have the same data available. As a result, mandatory pageflow parameters are mandatory in all user task forms within the pageflow. However, if this is not the desired behavior, explicitly associate the parameter with the user task in the pageflow and de-select the mandatory flag.

Resource Patterns and Work Distribution

This section describes patterns that are available to model how you want work to be distributed to resources. Resources are the people who carry out the work, and are represented in TIBCO Business Studio by participants. How these patterns are interpreted depends on your runtime environment, which may not support all the patterns.



The Workflow Patterns initiative (a joint effort of Eindhoven University of Technology and Queensland University of Technology) provides a conceptual basis for process technology. Their research identifies numerous patterns that can be supported by a workflow language or a business process modeling language. Many of these patterns are supported in TIBCO Business Studio. For more information about the Workflow Patterns initiative, see:

<http://www.workflowpatterns.com/index.php>

To support the modeling of workflow patterns, the following sections are available on the **Resource** tab in the Properties view:

- **Resources** (user and manual tasks)
- **Distribution Strategy** (message events and user tasks)
- **Piling** (user and manual tasks)
- **Separation of Duties** (user and manual tasks)
- **Retain Familiar** (user and manual tasks)

In addition to the patterns on the **Resource** tab, you can also use Chaining, which is configured on an embedded sub-process (see [Chained Execution on page 116](#)).

You can also access the Retain Familiar and Separation of Duties resource patterns by selecting tasks, right-clicking, and selecting **Resource Patterns**:



Resources

The participant in the Resources section is the same as the participant specified on the **General** tab, and specifies who or what completes the task (see [Creating Participants on page 180](#)). You can specify the **Initial Priority**, which indicates the relative urgency with which the item should be completed.

Distribution Strategy

Set the distribution strategy for the task to offer or allocate the work to a user:



The exact method with which work items are offered and allocated may differ, depending on the runtime environment.

- **Offer To All** Select this option to specify that you want *all* users that match the participant definition to have the opportunity to accept or decline the work item. For example, if there is a claims handler organizational entity (such as a group), the work item is offered to all users in that group. Once a user opens the work item, it is allocated to them and removed from the work lists of other users in that group.
- **Offer To One** Select this option to specify that you want only *one* user that matches the participant definition to have the opportunity to accept or decline the work item. If the user declines the work item, it is offered to another user that matches the participant definition.
- **Allocate To One** Select this option to automatically allocate the work item to a user that matches the participant definition.

Piling

Specifying that a user task may be piled means that multiple instances of that user task in a user's work queue will be presented to the user in sequence (in preference to other work items).

To specify piling on a work item, select the **May be piled** checkbox and specify the maximum number of items to pile. For example:

▼ Piling...

May be piled ☒

Maximum items to pile:

Separation of Duties

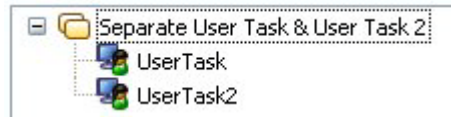
This pattern stipulates that you want specific tasks executed by different resources. For example, the resource that prepares a contract is different from the one who witnesses it. There are several ways to specify this pattern:

- On the right-click menu in the Process Editor
- On the **Resources** tab in the Properties view for the task
- On the **Task Groups** tab in the Properties view for the process

Right-Click Menu

Specify that tasks should be considered separate by using the right-click menu on the Process Editor as follows:

1. Select the user or manual tasks that you want performed by separate resources (use the Ctrl or Shift keys to select multiple tasks).
2. Right-click and select **Resource Patterns > Create a Separation of Duties Group**.
3. If you have not already done so, you must assign participants to the tasks.
4. The selected tasks are displayed on the **Resources** tab in the Properties view:



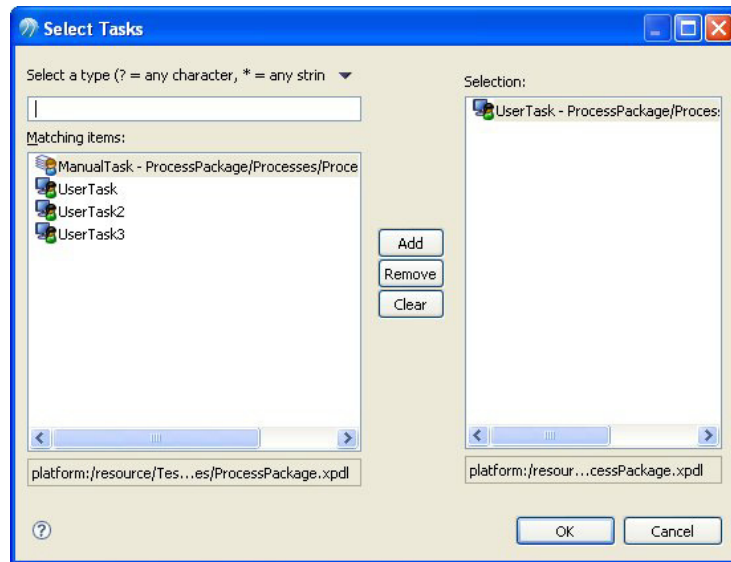
To see other task groups, click the **See all task groups(s) in the process** link on the **Resources** tab. This displays the **Task Groups** tab for the process. This shows task groups of which this task is a member.

Resources Tab

Specify that tasks should be considered separate on the **Resources** tab for the task as follows. If you have already set up a task group, the **Resources** tab shows the groups to which the selected task belongs.

1. Click the **Resources** tab and expand the **Separation of Duties** section.

- Click the  button to create a new task group. The **Select Tasks** dialog is displayed.



- Select the tasks that you want to include in the group and click **OK**.
- Once you have selected the task group members, you can also use the following buttons:
 -  Use this button to add or remove tasks to and from the group.
 -  Use this button to either delete tasks for the group or delete the group.



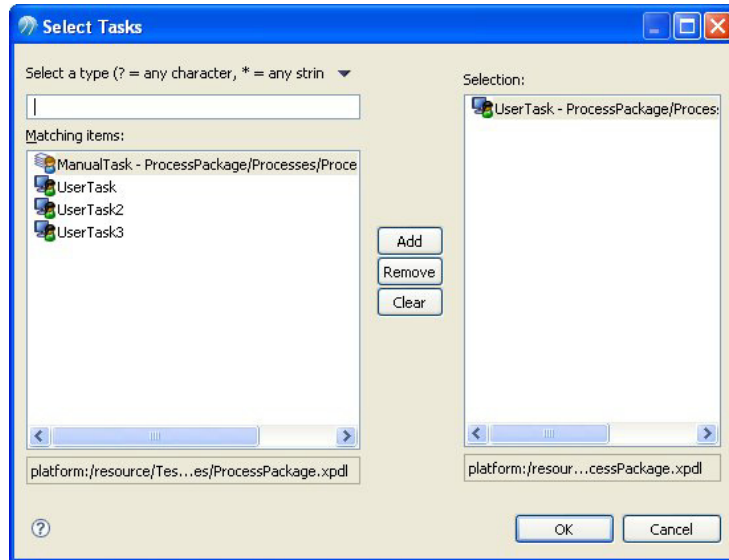
To see other task groups, click the **See all task groups(s) in the process** link on the **Resources** tab. This displays the **Task Groups** tab for the process. This shows task groups of which this task is a member.

Task Groups Tab

Specify that tasks should be considered separate on the **Task Groups** tab for the process as follows. If you have already set up a task group, the **Task Groups** tab shows the groups contained in the selected process.

- Select the process in the Project Explorer and click the **Task Groups** tab.

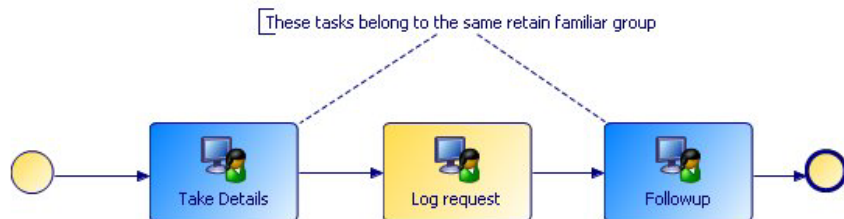
- Click the  button to create a new task group. The **Select Tasks** dialog is displayed.



- Select the tasks that you want to include in the group and click **OK**.
- Once you have selected the task group members, you can also use the following buttons:
 -  Use this button to add or remove tasks to and from the group.
 -  Use this button to either delete tasks for the group or delete the group.

Retain Familiar

This pattern stipulates that you want a specific task to be executed by the same resource that executed a previous task in the same process instance. For example, the resource that handles the initial customer contact is the same one that handles the follow-up call. Consider the following process:



In this example, all three tasks are assigned to the same organization unit. The **Take Details** task is allocated to a resource from the organization unit. Because the **Followup** task is in the same Retain Familiar group as the **Take Details** task, they will be allocated to the same resource as well.



This pattern specifies that you would ideally like tasks to be executed by the same resource. However, this is not always possible in the runtime environment. For example, it may be that by the time the later tasks in a retain familiar group are reached, the original participant is unavailable or not in the participant set. In such cases, the work item is delivered as normal according to the distribution strategy.

Where there are parallel threads it is possible for the second task to become active before the other has completed in which case it would be impossible to retain familiar. This situation is disallowed through validation within Business Studio such that the two activities can be guaranteed to be sequential.

There are several ways to specify this pattern:

- On the right-click menu in the Process Editor
- On the **Resources** tab in the Properties view for the task
- On the **Task Groups** tab in the Properties view for the process

Right-Click Menu

Specify tasks should be part of a Retain Familiar group using the right-click menu on the Process Editor as follows:

1. Select the user or manual tasks that you want to include in the group (use the Ctrl or Shift keys to select multiple tasks).
2. Right-click and select **Resource Patterns > Create a Retain Familiar Group**.
3. If you have not already done so, you must assign participants to the tasks.

4. The selected tasks are displayed on the **Resources** tab in the Properties view:

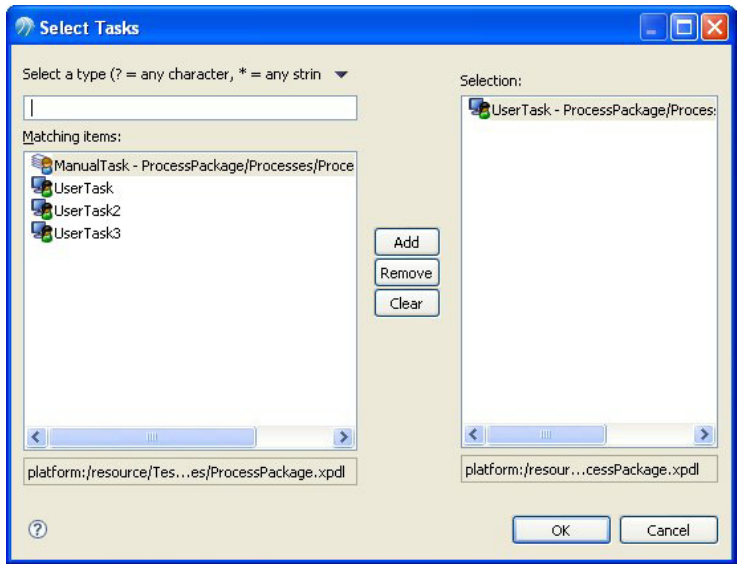


To see other task groups, click the **See all task groups(s) in the process** link on the **Resources** tab. This displays the **Task Groups** tab for the process. This shows task groups of which this task is a member.

Resources Tab

Specify that tasks should be part of a Retain Familiar group on the **Resources** tab for the task as follows. If you have already set up a task group, the **Resources** tab shows the groups to which the selected task belongs.

1. Click the **Resources** tab and expand the **Retain Familiar** section.
2. Click the  button to create a new task group. The **Select Tasks** dialog is displayed.



3. Select the tasks that you want to include in the group and click **OK**.

4. Once you have selected the task group members, you can also use the following buttons:
 -  Use this button to add or remove tasks to and from the group.
 -  Use this button to either delete tasks for the group or delete the group.

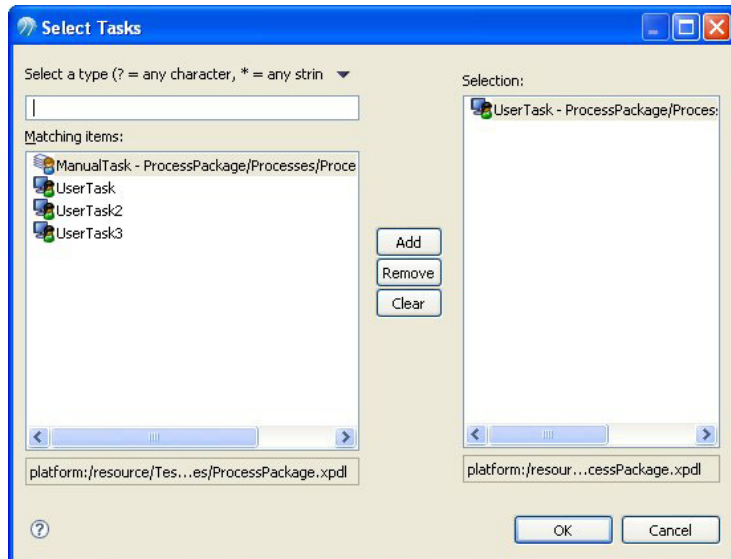


To see other task groups, click the **See all task groups(s) in the process** link on the **Resources** tab. This displays the **Task Groups** tab for the process. This shows task groups of which this task is a member.

Task Groups Tab

Specify that tasks should be considered separate on the **Task Groups** tab for the process as follows. If you have already set up a task group, the **Task Groups** tab shows the groups contained in the selected process.

1. Select the process in the Project Explorer and click the **Task Groups** tab.
2. Click the  button to create a new task group. The **Select Tasks** dialog is displayed.



3. Select the tasks that you want to include in the group and click **OK**.
4. Once you have selected the task group members, you can also use the following buttons:
 -  Use this button to add or remove tasks to and from the group.
 -  Use this button to either delete tasks for the group or delete the group.

Chained Execution

This resource pattern specifies the intention that the workflow engine should automatically start the next work item in a case once the previous one has completed. For more information, see the workflow patterns web site:

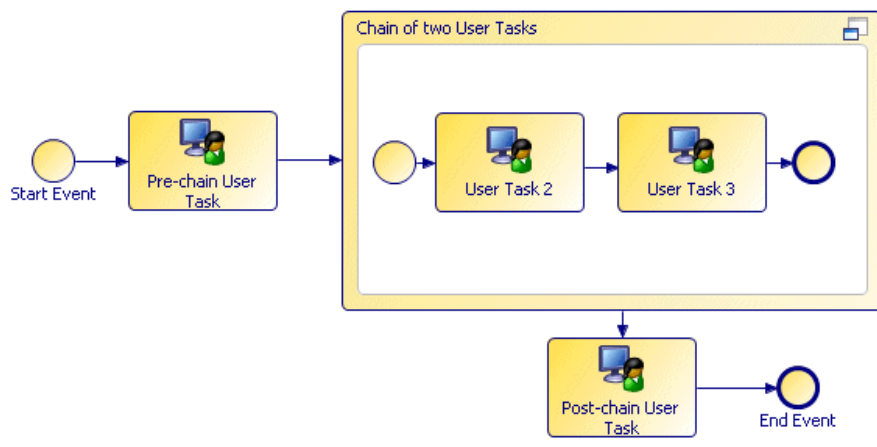
<http://workflowpatterns.com/patterns/resource/autostart/wrp39.php>

This has the effect of a resource being allocated sequential work items within a process instance, and when a work item is completed, the next task is immediately initiated. This keeps the resource constantly processing a given process instance.

To provide chained execution, configure an embedded sub-process as follows:



You can refactor existing tasks into an embedded sub-process as described in [Creating a New Embedded Sub-Process on page 197](#).



In the Properties view for the embedded sub-process, select **Chained Execution** in the Properties view:

Properties Problems Fragments

Embedded Sub-Process

General Label: Embedded Sub-Process

Description Name: EmbeddedSubProcess

Scripts

Appearance Activity Markers: ☐ Standard Loop ☐ Multiple Instance Loop ☐ Ad-Hoc

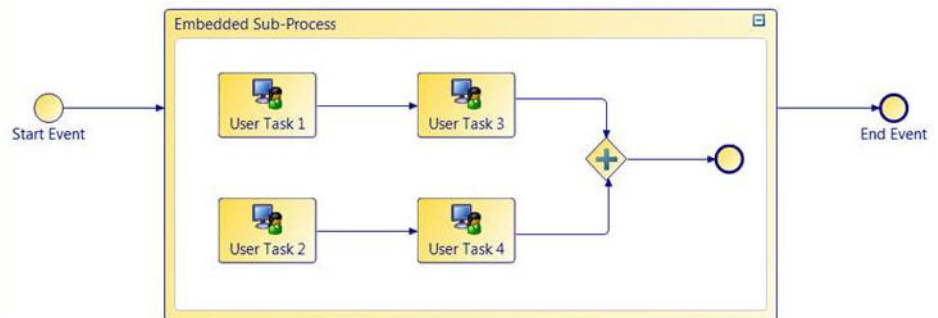
Extended Participants: - not set - ... Clear

Advanced Activity Type: Embedded Sub-Process

☐ Is A Transaction
☐ Start A New Trans
☒ Chained Execution

Multiple Parallel Paths in a Chaining Group

When multiple parallel paths exist in a chaining group, items will be chained in the order they are scheduled, and not the order of process flow. For example:



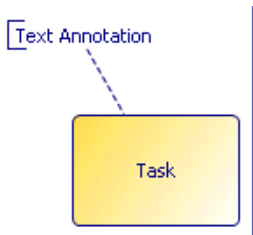
In the example above, User Task 1 or User Task 2 would be performed first. Both would appear in the relevant user's work list. If User Task 1 was opened first then on completion User Task 2 would be performed (it would be the only user task in the chained group available at that point). It is likely that whilst User Task 2 is completed that User Task 3 would be scheduled. This would be performed and then User Task 4 would be performed last. The user tasks would therefore be performed in scheduled order and not according to the connections between the user tasks.

Making Processes Easier to Follow

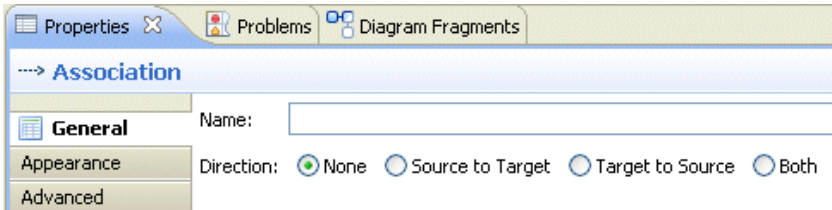
Data objects, text annotations and groups, all of which do not affect the sequence or message flow of the process can be used to make a process easier to follow.

Associations

An association is a connection from a data object or text annotation to a flow object (for example, an activity) used to make a process more readable:



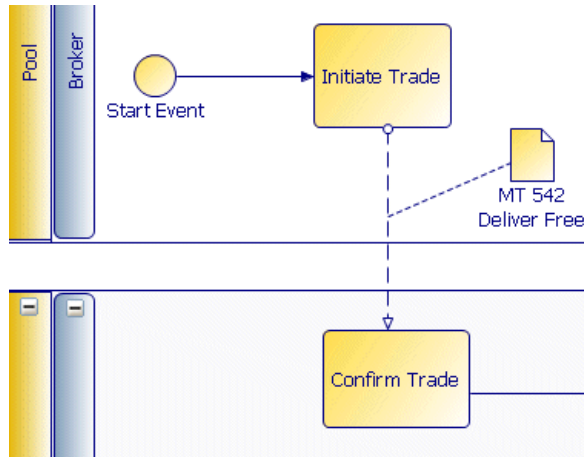
You can change the direction of the association in the Properties view:



Data Objects

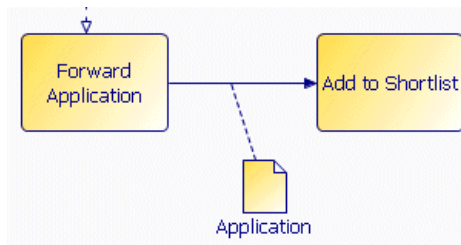
A data object is used for informational purposes to show how a document or other data relates to the process. It may be associated with a sequence flow or message flow, but it does not affect either flow.

Data objects are usually associated with flow objects with an association:



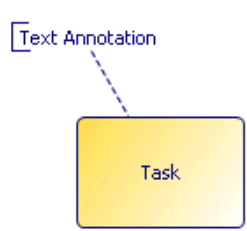
The data object in the preceding example should not be confused with the actual message being sent between the two pools.

A data object can also be associated with a sequence flow or other flow object:



Text Annotations

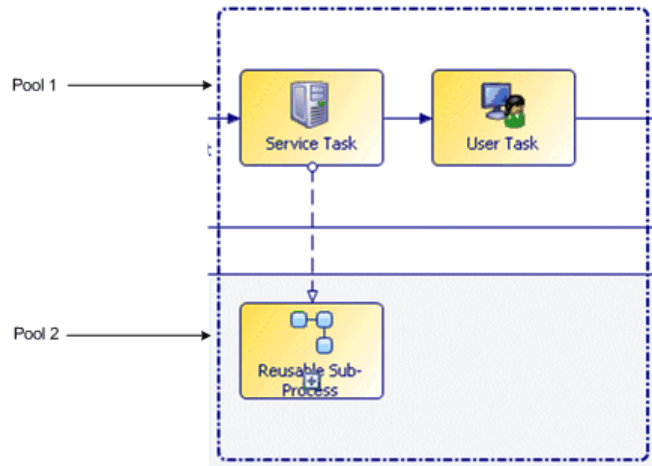
Text annotations serve to explain or clarify the process. They can be connected to flow objects (for example, tasks) using an association:



Unlike any text that you enter on the **Description** tab in the Properties view for an object, text annotations are displayed directly on the process.

Groups

Groups are used to indicate a relationship between elements of a process by enclosing them in a dashed line. A group can span lanes and pools. For example:



Groups are not preserved upon export because they have no meaning in the runtime environment.



Because the group is a separate layer that is overlaid on the process, it is not deleted when a lane or pool is deleted - even when the group is fully enclosed within the lane or pool.

Swimlanes

Swimlanes consist of [Lanes](#) and [Pools](#).

Lanes

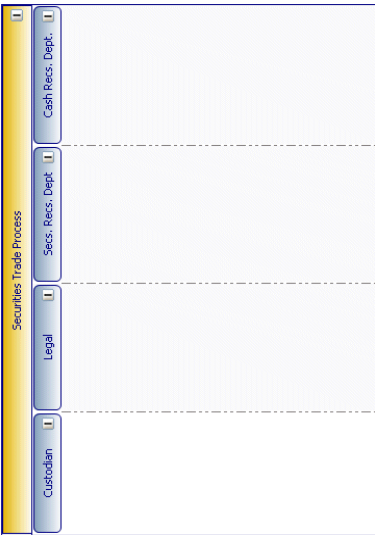
A lane is a subdivision of a pool that is used as a container for partitioning processes. How you use lanes is specific to your business. A lane can be **Normal** or **Closed**. The contents of a normal lane are visible. The contents of a closed lane are not visible and are used for "black-box" processes where you do not know the details of the contained processes.



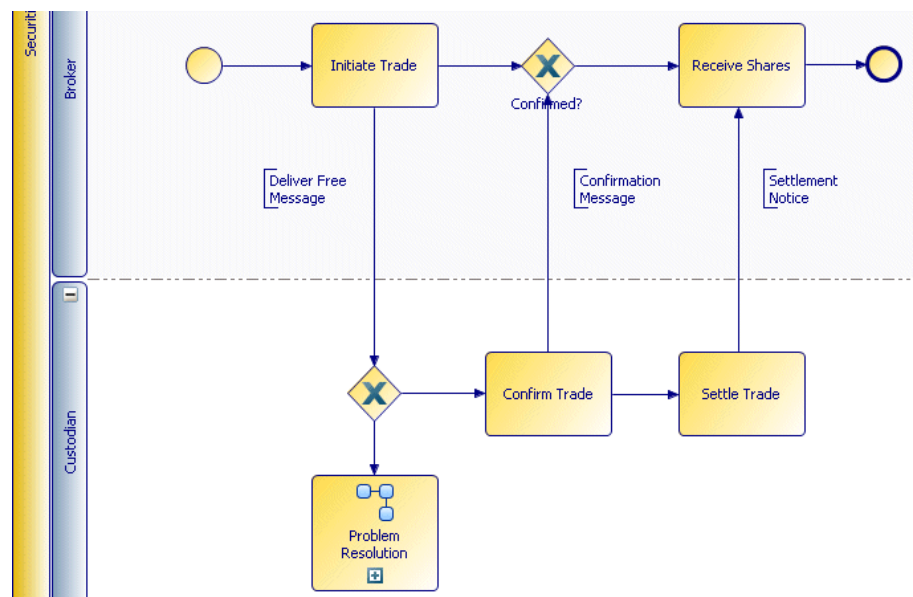
If you delete a lane in **TIBCO Business Studio**, all the objects in that lane are also deleted. If you do this inadvertently, press Ctrl+Z to restore the contents of the lane.

Examples

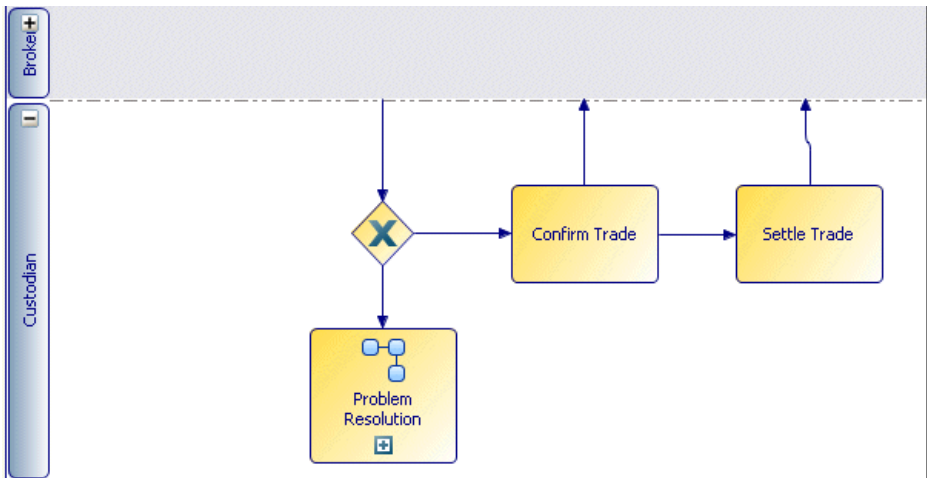
Suppose the back office of a financial institution has a process in which the Securities Reconciliations Department, the Cash Reconciliations Department, the Legal Department and a custodian are all involved. When defining this process, they can be represented by lanes within the pool:



Sequence flow can connect objects in different lanes:



The previous example shows a securities trade between a Broker and a Custodian. The same trade could be represented as follows:



This shows the Custodian lane in a Normal state and the Broker lane in a Closed state. The trade might be represented this way because it is documenting the Custodian’s part in the trade and the Custodian has no knowledge of the internal processes of the Broker.

Pools

A pool is used as a container for partitioning processes in ways that make sense for your business.

A pool is commonly used to document a process in a self-contained system. Typically the business analyst focusses on just one system or pool, but sometimes needs to show interactions with outside systems. For example, a customer places order and does not know how it is fulfilled in the closed supplier pool, but at some point in the process the customer pool receives a fulfillment response (message).

Pools are used in conjunction with lanes and are also related to message flow, which can also be used to show message flow between objects in different pools (see [Message Flows on page 126](#)).

Chapter 6 **Controlling Flow**

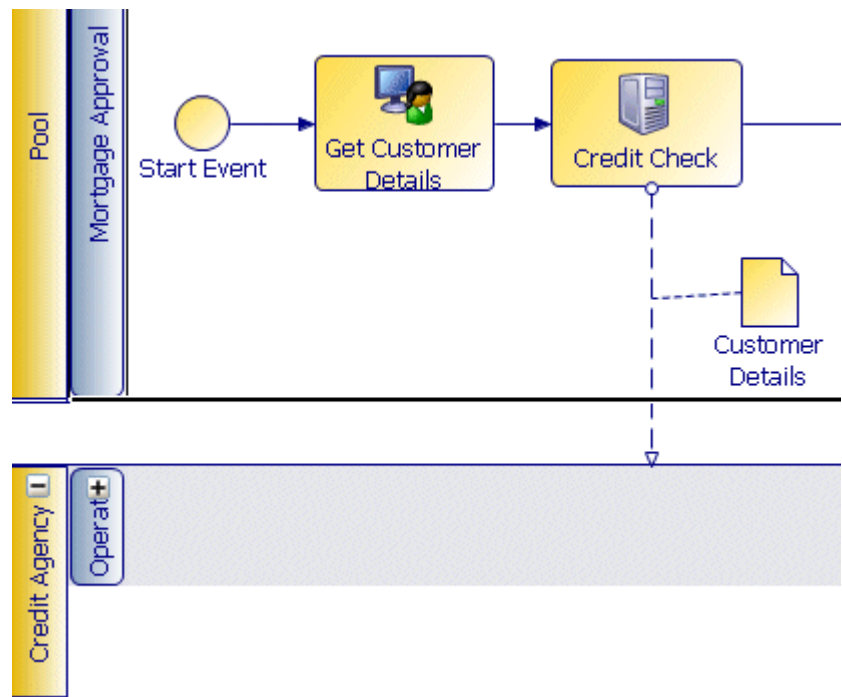
This section describes some of the common tasks that you perform using TIBCO Business Studio to control flow in a process.

Topics

- [Message Flows, page 126](#)
- [Sequence Flows, page 127](#)
- [Using Loops, page 128](#)
- [Gateways, page 134](#)

Message Flows

A message flow indicates the flow of messages between objects in separate pools or between pools. For example:



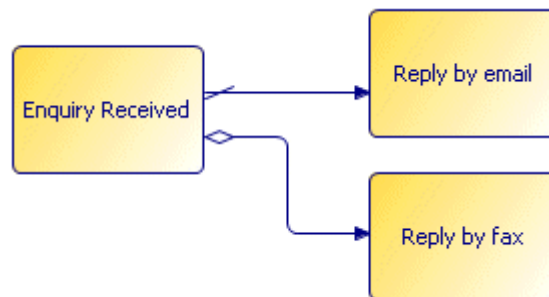
This process shows a mortgage approval process where the customer details are sent to the credit agency for approval using a message. In this case the pool and lane that represent the credit agency process are closed because we either do not care about or are not familiar with their internal processes.

Sequence Flows

Sequence flows indicate the order in which activities will be performed. You can set up sequence flows between flow objects (activities, events, and gateways). When you create a sequence flow, you can highlight it, right-click and select one of the following types of sequence flow:

-  Default Flow - Shows the default flow from a gateway or activity that will be taken if no conditional flow has its condition met.
-  Conditional Flow - Shows a flow that is only followed if the associated condition is met.
-  Uncontrolled Flow - Indicates a flow that does not have a condition associated with it (the default). This path will be taken in all cases.

Example



In this example, an enquiry is responded to either by email (the default), or depending on a condition, by fax.

Using Loops

TIBCO Business Studio supports standard and multi-instance loops as defined in BPMN.

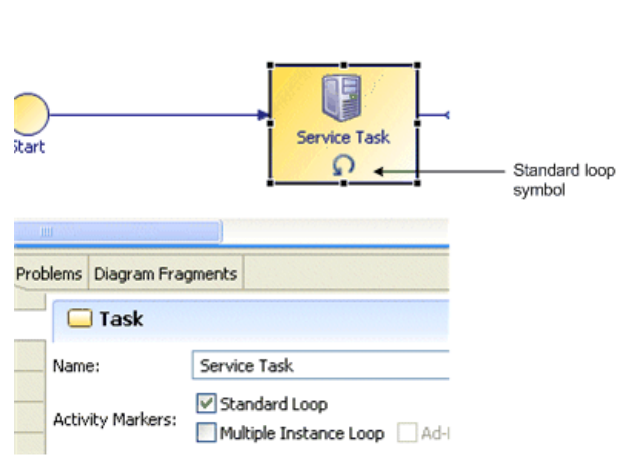
- **Standard loop** A standard loop consists of a Boolean expression that is evaluated either **before** (loop while condition is true) or **after** (loop until condition is true) each cycle of the loop.
- **Multi-instance loop** A multi-instance loop has an expression that evaluates to an integer, and is evaluated only once (before the activity is performed). The resulting integer specifies the number of times the activity should be performed.

You can select the type of loop applied to a task either in the Properties view for that task (on the **General** tab), or by right-clicking the task and selecting from the **Activity Markers** menu. The **Loops** tab in the Properties view for the task is where you specify the details of the loop.

This section describes how to create loops in TIBCO Business Studio, and how to implement the supported BPMN patterns.

Creating a Standard Loop

A standard loop is indicated by the following symbol:



Click the **Loops** tab to set the details of the loop:

The screenshot shows the 'Task' configuration window with the 'Loops' tab active. On the left, there is a sidebar with tabs: 'General', 'Description', 'Interface', 'Data Fields', and 'Loops' (which is highlighted with a blue icon). The main area is divided into two sections. The left section contains 'Evaluate Loop Condition:' with a dropdown menu showing 'Before (loop while condition is true)' and 'Loop Maximum:' with an empty text box. The right section contains 'Script Defined As:' with a dropdown menu showing 'Unspecified'.

Set the following parameters as appropriate for the loop you want to create:

- **Test Time** Select either **Before** or **After**. The **Before** option is equivalent to the programming construct "while": the expression is evaluated *before* the activity is performed, and therefore if the expression evaluates to False, the activity is not performed. The **After** option corresponds to the programming construct "Do: while": the expression is evaluated after the activity has been performed, guaranteeing that the activity is performed at least once.
- **Loop Maximum** Optionally specify an integer to control the maximum number of times the activity is performed. In the event of conflict between **Loop Maximum** and the result of the Javascript defined under **Script Defined As**, the **Loop Maximum** will override the script calculation.
- **Script Defined As** Select one of the following options:
 - **Free Text** Allows you to describe how you would like the loop to be tested if you prefer to leave the specific implementation of the loop expression to someone else.



A BPMN warning is generated if you leave the script area blank. To remove the warning, enter a text description of the intended script implementation.

- **JavaScript** Allows you to enter JavaScript that will be evaluated for the loop expression.
- **Unspecified** Allows you to specify that there is no expression or description for this loop.

Creating a Multi-Instance Loop

A parallel multi-instance loop is indicated by the following symbol (a sequential multi-instance loop uses the standard symbol):



Click the Loops tab to set the details of the loop:

The screenshot shows the 'Task' configuration window with the 'Loops' tab selected. On the left is a sidebar with tabs: General, Description, Interface, Data Fields, and Loops (which is active). The main area has two sections. The first section has 'Ordering:' set to 'Sequential' and 'Flow Condition:' with a dropdown menu open. The dropdown menu lists four options: 'All (perform outgoing flow when all instances complete)', 'One (perform outgoing flow once when first instance completes)', 'No Condition (perform outgoing flow for each completed instance)', and 'Complex (behavior defined by Complex Flow Condition)'. The second section is titled 'Scripts for Multi Instance:' and contains 'Loop Expression' set to 'Complex Flow Condition' and 'As:' set to 'Unspecified'.

Set the following parameters as appropriate for the loop you want to create:

Ordering

Select either **Sequential** or **Parallel**. The Sequential option causes the instances of the activity to occur in succession rather than at the same time. The Parallel option causes the instances of the activity to occur at the same time.

Flow Condition

This property can be used to mimic the functions of a gateway. Select one of the following options:

- **All (perform outgoing flow when all instances complete)** The outgoing flow is processed once after all activity instances have completed.
- **One (perform outgoing flow once when first instance completes)** The outgoing flow is processed when the first activity instance completes (existing instances can still be completed but the outgoing flow is not processed when they complete).
- **No Condition (perform outgoing flow for each completed instance)** The outgoing flow is processed for every activity instance as it completes.
- **Complex (behavior defined by Complex Flow Condition)** The outgoing flow is processed for each instance for which the Complex Flow Condition tab evaluates to true.

Scripts for Multi-Instance

Select **Free Text** for the **Script Defined As** field if you want to describe how you would like the loop to be tested, and prefer to leave the specific implementation of the loop expression to someone else. You can also select **Unspecified** if there is currently no condition or description for the loop.



Depending on the destination environment you have selected, **JavaScript** may be available as an option. Use this option if you want to enter JavaScript that will be evaluated for the loop expression.

There are three tabs on which you can specify scripts:

- **Loop Expression** Specify either a script that evaluates to an integer or a description of the desired script.
- **Complex Flow Condition** This expression is evaluated if you selected Complex as the Flow Condition.
- **Additional Instances Expression** This expression is used for control-flow pattern WCP-15 (additional activity instances may be required at runtime) when you only need to add instances when the task is complete. If you need to add instances while the task is in progress, see [Adding Additional Instances to a Multi-Instance Loop Task \(whilst that task is in progress\) \(WCP-15\)](#).

The additional instances script is evaluated after the last instance completes. If the script evaluation calls for additional instances, they will be done and the script will be evaluated again once the last additional instance completes.



If an Additional Instances Expression is specified then this expression must eventually evaluate to zero so no more instances occur.

Adding Additional Instances to a Multi-Instance Loop Task (whilst that task is in progress) (WCP-15)

To add additional instances to a multi-instance loop task whilst that task is in progress, you need to execute a script somewhere on a parallel path whilst the loop task is active. The script can add instances to the loop task using an expression similar to the following:

```
Process.addActivityLoopAdditionalInstances ("OrderStock",1);
```

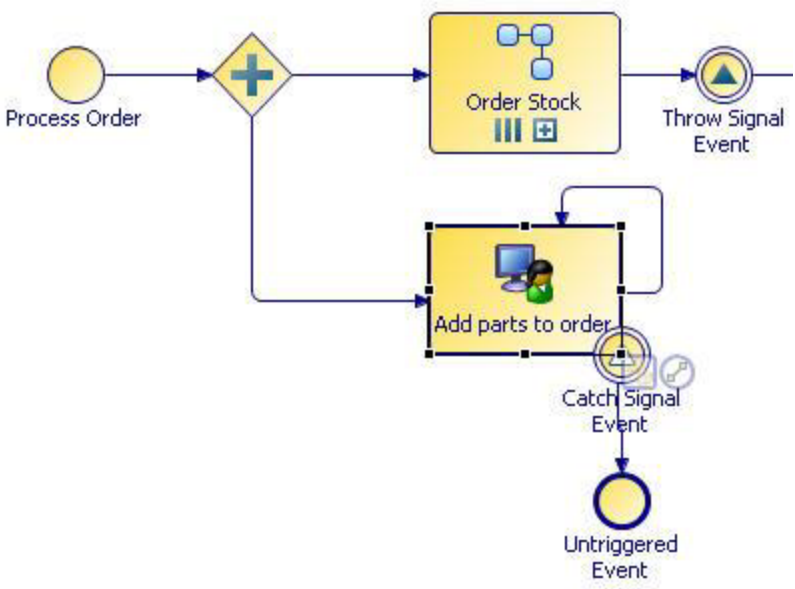
which contains a string for the name of the task (in this example, OrderStock), and an integer for the number of additional instances required (in this example, 1).



If you wish to add additional instance to a dynamic sub-process task, you need to pre-populate the runtime-identifier array with the name or names of the sub-process implementation that you wish to add, as well as any associated arrays used to provide input into the dynamic sub-process.

This can also be performed in the loop task itself **but only** in the task Complete script.

In this example, the task would order an additional item each time it was used.



▼ Process Manager Scripts

Initiate Script

Complete Script

Timeout Script

Cancel Script

Script executed by the process engine on completion of the activity.

Script Defined As:

JavaScript

Describe Completed Script:

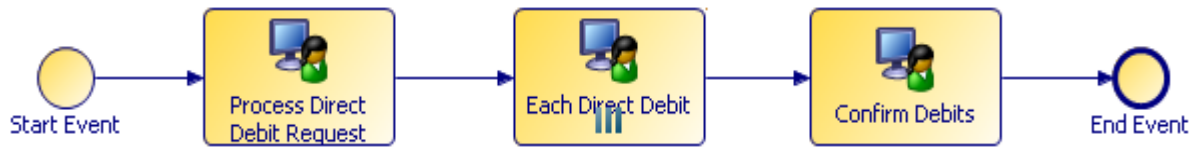
Process.addActivityLoopAdditionalInstances("OrderStock",1);

Example - Multiple Instances with Synchronization (WCP-13)

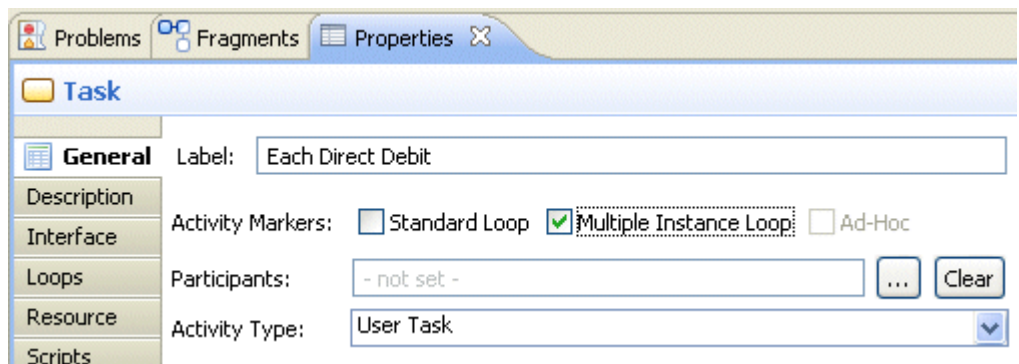
This workflow pattern specifies that multiple instances of an activity should be created, each instance being synchronized and able to run concurrently. Note that:

- The number of instances required is known and specified at design time.
- Each task instance has access to all the process data (data fields, parameters, and so on).
- All the instances must be complete before the next task is started.

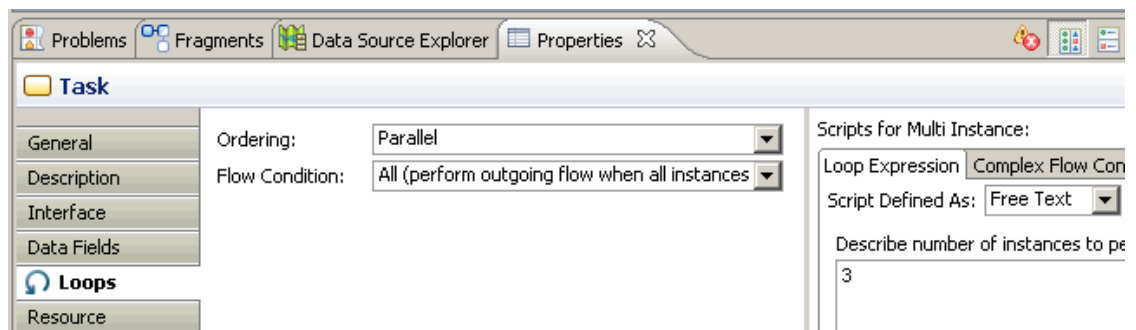
The following process has three activities:



There can be multiple instances of the Each Direct Debit task. This is indicated on the General tab of the Properties view:



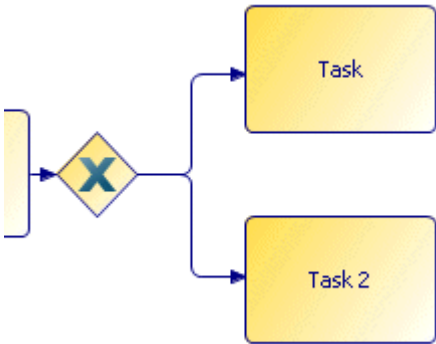
The details are specified on the Loops tab:



The **Parallel** ordering setting and **All** flow condition means that the activity instances will be performed at the same time and will be synchronized. The number of instances is set to 3 on the right side of the Properties view.

Gateways

Gateways are a control mechanism for the sequence flow in the process. They are represented by a diamond:

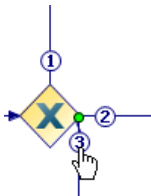


Although the gateway resembles a decision box in a flow chart, gateways provide a variety of behaviors besides conditional decisions.

As shown on the Properties view of a gateway, these are the different types of gateway that you can create.

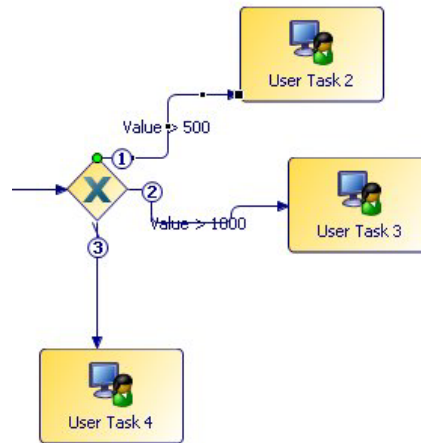
Order of Flow Evaluation

When a gateway has multiple sequence flow output, you can specify the order in which the outgoing sequence flow is processed. You can view the current order by highlighting one of the sequence flows and placing the pointer over the outline numbers that appear. For example:



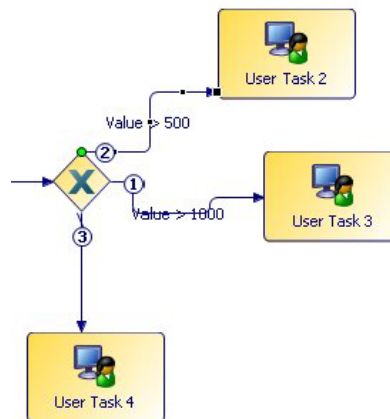
Gateways that are being deployed to iProcess cannot have multiple conditional output sequence flows.

This is especially significant for evaluating the conditions on sequence flows attached to exclusive gateways. For example:



In this case, the > 500 sequence flow is processed first. Assuming that the JavaScript conditions are set up as their labels imply, if the value is greater than 500, **User Task 2** will be executed. The > 1000 sequence flow and **User Task 3** will never be reached.

To change the order of evaluation, drag the numbers that appear on the sequence flows. For example, dragging the 2 and dropping it onto the 1 changes the order of sequence flow evaluation as follows:



Regardless of the selected order of evaluation, conditional sequence flows are always evaluated *before* default sequence flows.

Exclusive (XOR)

In an exclusive gateway, there are several paths through which the process can continue, but only one is actually chosen when the process is run. There are two types of exclusive gateway:

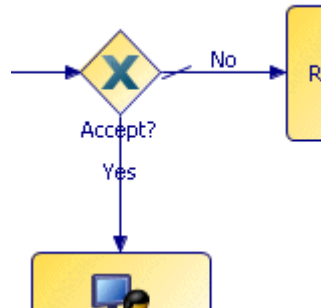
- **Exclusive (Data)** The sequence flow is chosen based on an expression using data from the process. This type of gateway is indicated in the process as follows:



- **Exclusive (Event)** The sequence flow is chosen based upon an external event (for example, a JMS message). This type of gateway is indicated in the process as follows:



The following shows a typical XOR (data) gateway:



There is one uncontrolled input sequence flow to the gateway, and conditional and default output sequence flows.



An XOR (data) gateway displays an **X** as a visual cue to the gateway type. However, because the display of the **X** is not required by BPMN, you can disable the display of the **X** in the Property view for the gateway by deselecting **Show "X" Marker**.

Inclusive (OR)

An inclusive gateway looks like this:



In an inclusive gateway used for branching, each output Sequence Flow is independently evaluated according to an expression. This means that anywhere from zero to the maximum output sequence flows can be taken. In practice, you should either provide a default sequence flow or ensure that at least one sequence flow evaluates to True.

When used to merge flow, any upstream sequence flows are synchronized, but the gateway does not wait for all sequence flows.

Complex

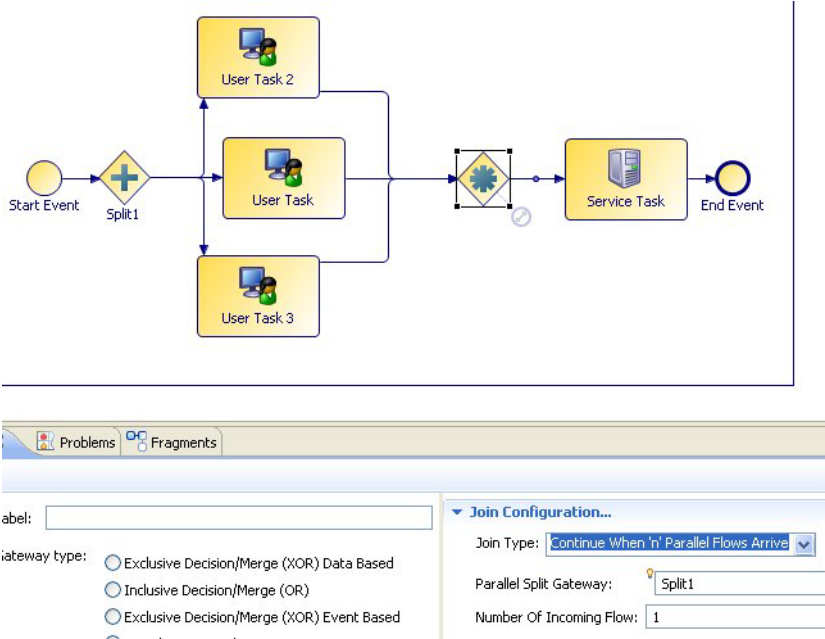
A complex gateway is used to fork or merge depending on how an expression evaluates. When used as a decision, the expression determines which of the outgoing sequence flow are chosen for the process to continue.

When used to merge flow, the expression determines which of the incoming sequence flows is required for the process to continue. This type of gateway is indicated in the process as follows:



Join Configuration

Although TIBCO Business Studio does not provide for entering an expression on a complex gateway, there is a **Join Configuration** section in the Properties view. This allows you to specify how many incoming sequence flows are received before flow continues. For example:



There are three sequence flows going into the complex gateway. On the properties of the gateway, the **Continue When 'n' Parallel Flows Arrive** join type is selected. The parallel gate way is specified indicating that this complex gateway is handling an earlier parallel split (named **Split1**), and that flow should continue when only one of the sequence flows reaches the complex gateway.

Parallel (AND)

A parallel gateway is used to fork or merge several parallel paths (synchronization). When several sequence flows enter a parallel gateway, the process flow waits until all arrive at the gateway before continuing. This type of gateway is indicated in the process as follows:



Chapter 7 **Working with Events**

This section describes some of the common tasks that you perform using BPMN events.

Topics

- [About Events, page 140](#)
- [Start Events, page 143](#)
- [Intermediate Events, page 144](#)
- [End Events, page 146](#)
- [Configuring Timer Event Scripts, page 147](#)
- [Using Throw and Catch Signal Events, page 149](#)
- [Using Throw and Catch Error Events, page 151](#)
- [Configuring Compensation Events, page 157](#)
- [Implementing Message Events, page 158](#)

About Events

An Event in a process is something that happens that affects the sequence or timing of activities in a process, for example the receipt of a message. There are three main types of Event: Start, Intermediate and End.

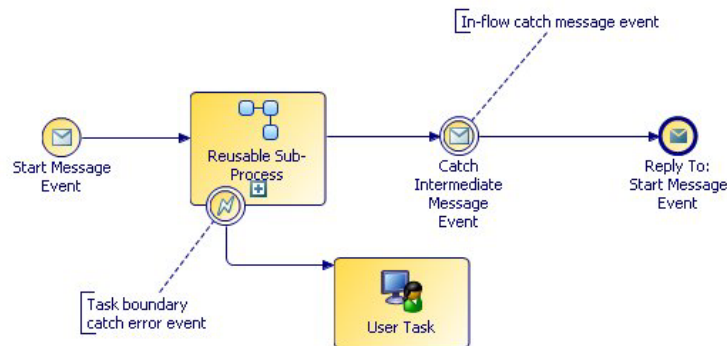
- Start events are used to indicate the start of a process or to control how a process is started (or triggered).
- Intermediate events can be placed after start events and before end events to indicate messages, exceptions, compensation, and so on. Most types of intermediate catch events can also be placed on the border of an activity.
- End events indicate the end of a flow or branch. End events are optional, however if a process contains a start event, it must contain an end event.

Throw and Catch Events

An event that is located in sequence flow "throws" an event that can be "caught" by a catch event. Additionally, throw events are distinguished from catch events on a diagram by having their symbol colored in:



Catch intermediate events can be placed in flow, or on a task border. An in-flow catch event halts the flow until the event is triggered. A catch event attached to a task border usually cancels the task when the event is triggered.



References Between Throw and Catch Events

As an alternative to creating references between throw and catch events in the Properties view, TIBCO Business Studio provides graphical tools called gadgets that allow you to easily create references. The example in this section shows throw and catch signal events, but the same procedure applies to all throw and catch events.

You can designate throw and catch event pairs on the diagram as follows:

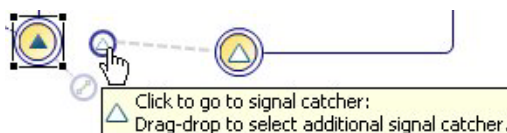
1. Place the throw and catch events on the diagram.
2. Select the throw event.



3. Drag the **Link Event** gadget to the desired catch event.



4. Once you link a throw and catch event, you can click the **Link Event** gadget to go to the catch event, or drag the **Link Event** gadget to select additional catch events:



Triggers and Results

An event *trigger* defines the cause for the event (for example, an error elsewhere in the Process). An end event *result* indicates the consequence of a sequence flow ending (for example, the sending of a message).

Triggers and results can either be "thrown" by events, or "caught" by other events as described.

Summary

The following table summarizes the BPMN events that are available to you on the palette in TIBCO Business Studio:

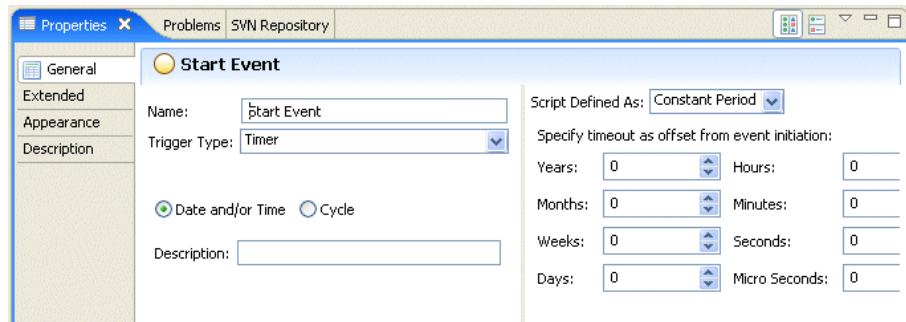
Table 2 Summary of Event Types

Event Type	Start	Intermediate	throw	End	Description
None					No specific trigger for the event (for example to start the process as a sub-process).
Message					Message is sent to do one of the following: start the process, resume the process, signify the end of the process.
Timer					Event is triggered at a specific date/date or regular time interval (time cycle).
Error					Ends a sub-process with an error. On a task boundary, either catches the specified error or any error if no error code is specified.
Cancel					Ends a transactional sub-process. On a task boundary, catches the cancel event thrown from within the sub-process.
Compensation					Either throws or catches a call for compensation. Used to process compensating activities for previously executed tasks.
Conditional					Triggered based on the evaluation of conditions.
Link					Creates a "go to" or "off page connector" to break up a process for better legibility.
Signal					Broadcasts or catches signals.
Multiple					Indicates that one of several possible triggers are to be thrown or caught.
Terminate					Ends the process and all activities within without compensation or error handling.

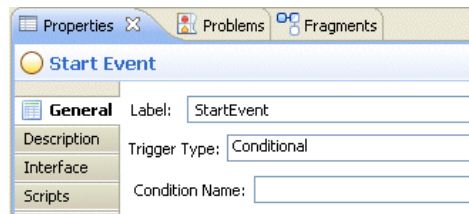
Start Events

Start events can be used to indicate the start of a process (they are optional). Different types of start events can control how a process is started (or triggered). All start events are "catch" events:

- **None** There is no specific trigger to start the Process.
- **Message** The process is started upon receipt of a message from an external source. This can be implemented using a web service using the Solution Design capability (see [Implementing Message Events on page 158](#)).
- **Timer** The process is started at a specific date/time or at a regular interval (time cycle):



- **Conditional** The Process is started by the evaluation of a condition:



- **Multiple** There can be several possible triggers for the start of the process, which are specified in the Properties view of the start event.
- **Signal** The start of the process is triggered by the receipt of a signal.

Intermediate Events

Intermediate events can be placed after start events and before end events to indicate messages, exceptions, compensation, and so on. Most types of intermediate events can also be placed on the border of an activity, where they are considered "catch" events. You can use the following types of Intermediate events:



BPMN imposes some restrictions on the placement of intermediate events. For example:

- Intermediate events of type **None** and **Link** cannot be placed on the boundary of a task.
- Intermediate events of type **Cancel** and **Multiple** cannot be placed in sequence flow.

TIBCO Business Studio allows you to place any type of intermediate event on the boundary of a task or in sequence flow; however any invalid constructions are reported in the Problems view.

- **None** Indicates an unspecified change in the process.
- **Throw/Catch Message** Either stops the flow of the process pending the receipt of a message (catch), or sends a message and resumes flow (throw). This can be implemented using a web service using the Solution Design capability (see [Implementing Message Events on page 158](#)).
- **Timer** The event is triggered at a specific date/time or at a regular interval (time cycle). When placed on the boundary of a task, a timer event defines a deadline for the task. In the Properties view for the event, there are two options you can select (**Withdraw Task on Timeout** and **Continue Task on Timeout**).



The timer deadline uses the appropriate system calendar (specific to a destination environment) to calculate when the deadline will expire. For more information about the system calendar, see the destination-specific implementation guide.

- **Conditional** The event is triggered based on the evaluation of a condition.
- **Throw/Catch Link** Indicates a connection from one or more throw link intermediate events to a catch link intermediate event in the same parent process. This can be thought of as a "go to" or "off page connector" that you can use to break up a process for better legibility.
- **Throw/Catch Signal** Broadcasts or catches a signal. A throw signal event is assigned a default signal name (**signaln**).

- **Throw/Catch Multiple** Indicates that there can be several possible triggers for the event.
- **Catch Error** Attached to a task boundary to end a sub-process with an error. Either catches the specified error, or catches any error if no specific error is specified.
- **Throw/Catch Compensation** Used to process compensating activities for previously executed tasks:
 - If located in sequence flow of the process, this event throws a call for compensation.
 - If attached to the boundary of an activity, this event catches a named compensation call.
- **Catch Cancel** This type of intermediate event is used on the boundary of a transaction sub-process. It is triggered if a cancel end event is reached within the transaction sub-process or if a transaction protocol “cancel” message is received while the transaction is being performed.

End Events

An end event indicates when the process has completed. They are optional, however if a process contains a start event, it must contain an end event. End events have different types that indicate different results upon completion of the process. All end events are "throw" events:

- **None** There is no specific end result to the process.
- **Message** Indicates that a message is sent at the end of the process. The message can either be a response to the start message, or a different message. This can be implemented using a web service using the Solution Design capability (see [Implementing Message Events on page 158](#)).
- **Signal** Indicates that a signal is broadcast at the end of the process. A signal end event broadcasts a default signal name (**signaln**).
- **Multiple** Indicates that there is more than one result that will occur when the process ends.
- **Error** Indicates that the specified error is generated at the conclusion of the sub-process.
- **Compensation** Indicates that a compensation is necessary. For more information, see the BPMN specification at <http://www.bpmn.org>.
- **Cancel** Used within a transaction sub-process to trigger a cancel intermediate event attached to the sub-process boundary.
- **Terminate** Ends all activities in the process immediately without compensation or events.

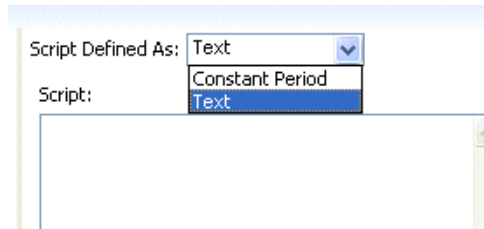
Configuring Timer Event Scripts

You can specify that a script is executed for a start timer event or an intermediate timer event (either inflow or on a task boundary).

For example, the following intermediate timer event has been placed on a task boundary:



In the Properties view, you can choose a script type from the **Script Defined As** list:



Depending on the destination environments selected, there can be other script types in the **Script Defined As** list.

Select one of the script types:

- **Free Text** If you want to leave the implementation of the script for the solution designer, you can select Text and use this area provided to describe the desired behavior for the script.
- **Constant Period** This allows you to specify the timeout period after the event is initiated using the following time units:

Script Defined As: Constant Period

Specify timeout as offset from event initiation:

Years:	0	Hours:	0
Months:	0	Minutes:	0
Weeks:	0	Seconds:	0
Days:	0	Micro Seconds:	0

- **JavaScript** This script type allows you to enter JavaScript statements in the space provided. For more information, see the appropriate implementation guide.

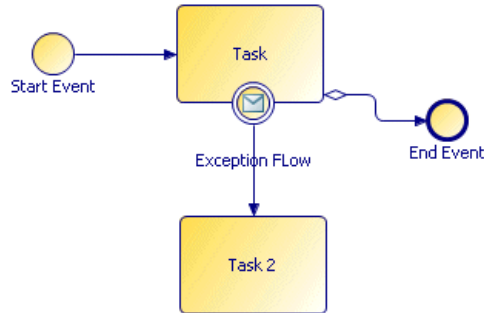


The JavaScript script type is not available in the Business Analysis capability (see [Capabilities and Perspectives on page 13](#)).

- **Unspecified** This means that no description or script is supplied.

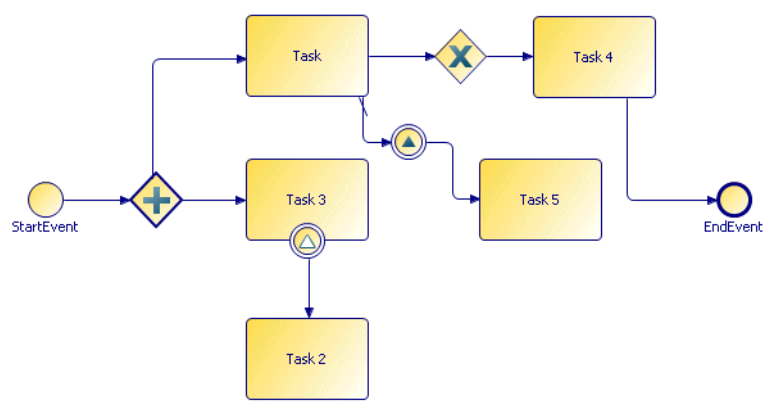
Using Throw and Catch Signal Events

In BPMN, when an event of any type on the boundary of a task fires, the task is cancelled and the exception flow from that event is followed. For example:



In this example, if the message event is fired, the task to which it is attached is cancelled and the sequence flow to Task 2 is followed.

With a throw/catch signal event pair, an *inflow* signal event broadcasts a signal, and if there is an active task with a signal event on the task boundary, that event "catches" the signal, cancelling the task and following the exception flow. For example:



In this example, if the default flow for the XOR gateway is taken, the inflow signal event broadcasts a signal. If the task with the signal event on its boundary is active, it will catch the signal, cancel the task and proceed with Task 2. If the task is *not* active, no cancellation occurs. This is useful when *either* Task 5 or Task must be completed.

You must specify a signal name with a throw signal event. With a catch signal event, a signal name is optional:

- If a catch signal event has no signal name specified, it catches any thrown signals in the process (while the task it is attached to is active).
- If a catch signal event has a signal name specified, it only catches signal events that throw that signal.



When you specify signal names in the Properties view for signal events, you can use content assist. This means for example, that if you have entered **SIGNAL1** for a throw signal event, when you specify the signal name for the catch signal event you can press Ctrl+Space and the available signal names are displayed from which you can select one.

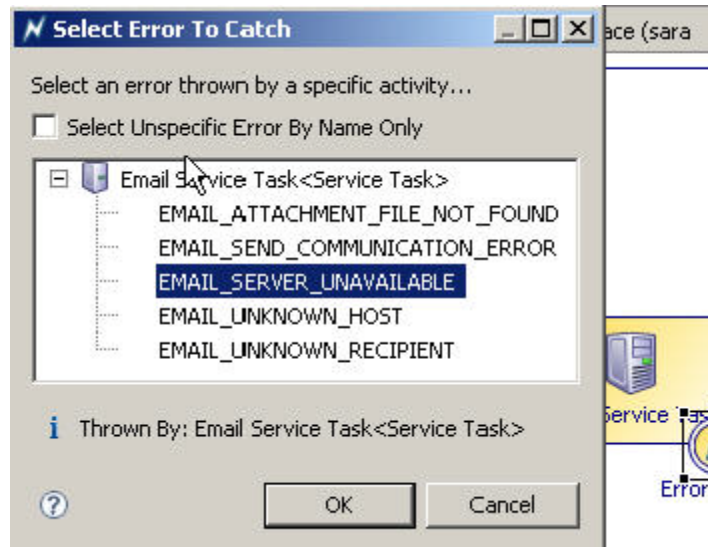
A signal end event functions just like any other inflow signal event, except that when one fires, if there are more tasks that need to complete as the result of exception flow, the outstanding tasks complete; if there are no more tasks that need to be completed, the process ends.

You can also use in-flow catch signal events to pause flow until the specified (or any signal) is broadcast.

Using Throw and Catch Error Events

An error event attached to a task boundary can be set to catch any error, or errors thrown by the task to which it is attached. For example:

- when attached to a web service task, you can configure the catch error event to catch any WSDL fault associated with the web service operation that is invoked by the service task.
- when attached to an email service task, you can configure the catch error event to catch emails when the email server is unavailable:

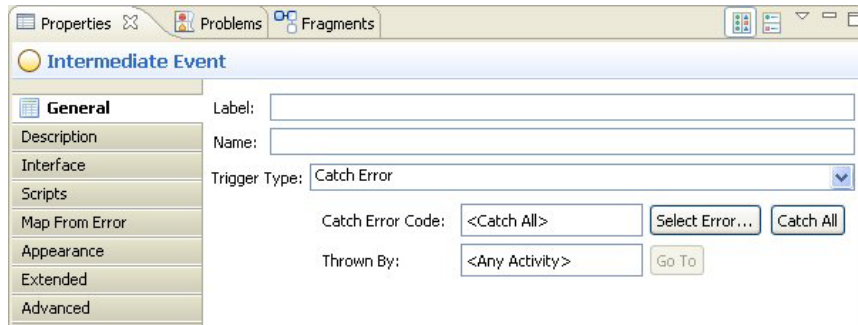


When attached to a reusable or embedded sub-process task, the event can catch errors thrown by any of the following:

- Activities within the sub-process.
- End error events
- Process interface error events (if the reusable sub-process task references a process interface rather than a process)
- Any other error throwing activity executed within the sub-process (including its sub-process tasks). This includes activities whose errors cannot be caught directly by attaching the error event (for example, throw message intermediate event and end event that invokes a one way message).

Configuring Error Events

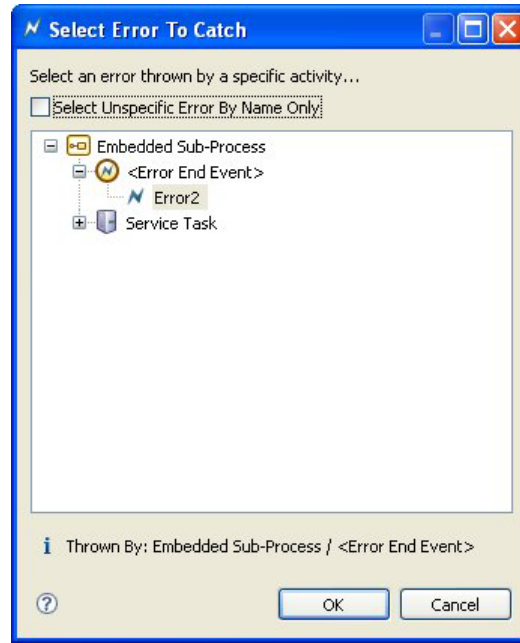
Error events are configured in the Properties view:



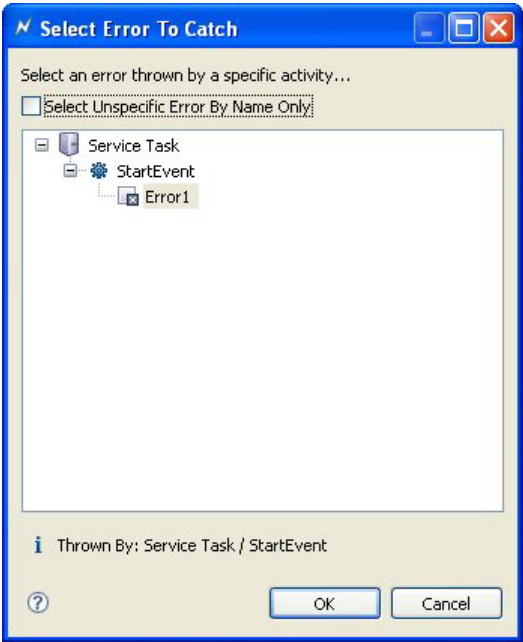
For a catch error event, you can do the following:

- **Catch All Errors** (*default*) Catches any error thrown by any event.
- **Catch Named Errors** To do this, click **Select Error** and select the **Select Unspecific Error By Name Only** checkbox. The dialog lists the catchable error codes. Select the error codes that you want to catch. This configures the event to catch the selected error from any activity.
- **Catch Specific Errors from Specific Activities** Click **Select Error** and deselect the **Select Unspecific Error By Name Only** checkbox. The dialog lists events

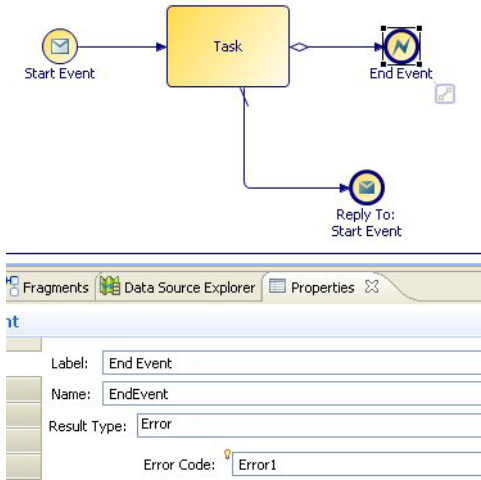
and activities that throw error codes. Expand the desired event or activity to catch a specific error thrown by a specific activity. For example:



- **Catch Specific WSDL Fault** WSDL faults are caught in the same ways as other errors (either named WSDL faults, or specific errors from a specific service task). For example:



In this example, the catch error event is attached to the service task. The error is thrown by a process in a different project (see [Creating References in the Project Properties on page 52](#)).



Fixing Invalid Error Events

Suppose that after configuring a catch error event, the event that throws the error code is deleted. This would invalidate the catch error event as shown in the Properties view:



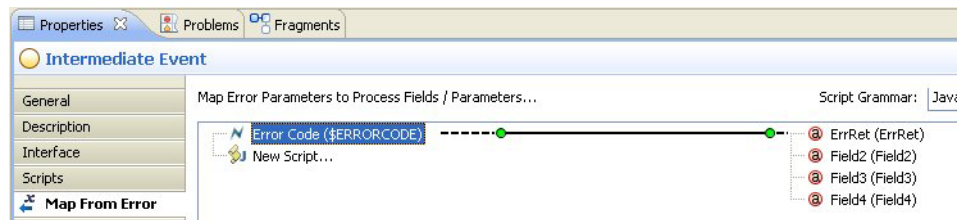
Other causes of invalid error events include:

- The event is detached from the task.
- The activity no longer throws the given error.
- The activity that throws a specific selected error is no longer processed as a result of the task the to which the event is attached.

To fix the error event, click **Select Error**. If possible TIBCO Business Studio identifies the closest match to the originally selected error code. Select an error code and click **OK**.

Mapping Error Data

Click the **Map From Error** tab to map error parameters to process data. For example:



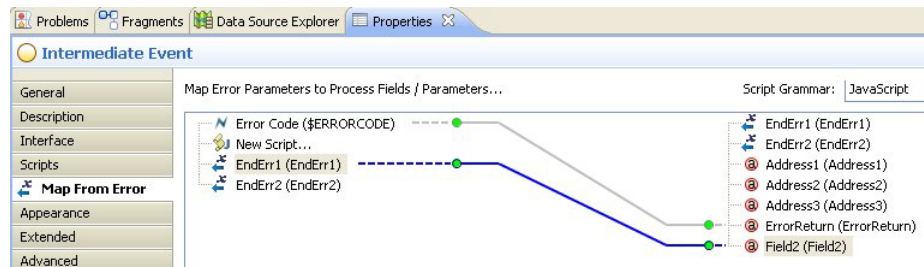
The process data available on the right side of the tab is either all data or a subset of data as specified on the selection on **Interface** tab.

The content of the left side of the **Map From Error** tab depends on the configuration of the **General** tab:

- **Catch All / Catch Named Errors** Only the automatically provided error code can be used for mapping as shown in the previous example. Typically, this error code is mapped to a process text data field or parameter for display to

the user. At run time, the text data field or parameters is populated with the error code name when the error is caught.

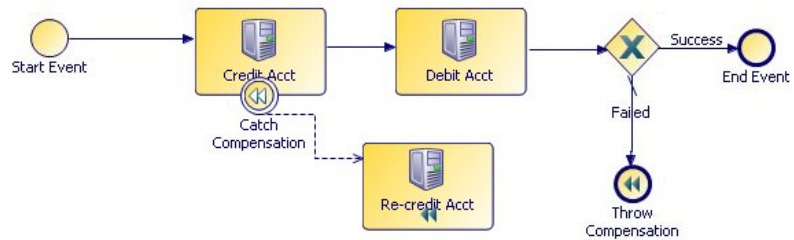
- **Catch Specific Error Thrown By Sub-Process End Error Event** All parameters with a mode of **Out** or **In/Out** that are associated with the throw error end event (on the **Interface** tab) are displayed for mapping:



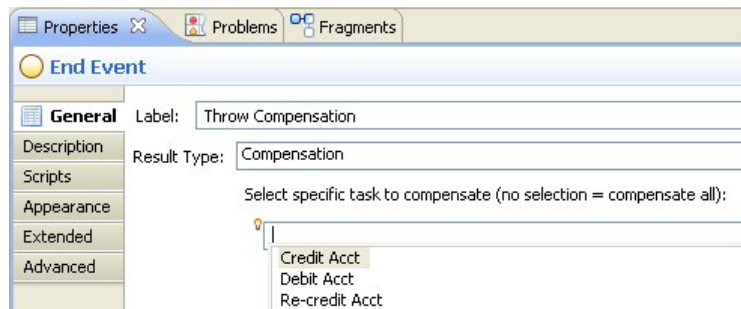
- **Catch Specific WSDL Fault** WSDL fault message parameters are displayed on the left side and can be mapped as normal.

Configuring Compensation Events

Compensation events are used to reverse the effect of previously executed tasks. For example:



In this example, if the credit/debit fails, a compensation event is thrown. The target of the event is the **Credit Acct** task, which catches the error and proceeds with the **Re-credit Acct** task. In the Properties view for the **Throw Compensation** end event, you can press Ctrl+Space to see a list of tasks from which you can select the target of the compensation. Alternatively, you can compensate all previously executed tasks by not specifying a target for the compensation:



Implementing Message Events

TIBCO Business Studio supports the following Message Exchange Patterns (MEPs):

MEP	Description	How to Model
Receive one-way request from external application	The process receives a message from an external application and does not send a response.	- Start message event (if the process is started upon receipt of the message) - Intermediate catch message event - Receive task See Receiving a One-Way Request Operation on page 159 and Receive Tasks on page 94.
Send one-way request to external application	The process sends a message to an external application but expects no response.	- End message event - Intermediate throw message event - Send task See Sending a One-Way Request Operation on page 160 and Send Tasks on page 94.
Receive request-response message from an external application	The process receives a request response message from the external application, and replies to the request.	- Start message event - Intermediate catch message event, or - Receive task <i>paired with</i> - Message end event - Intermediate throw message event, or - Send Task See Receiving a Request-Response Operation on page 160 and Request Response Operation with Send and Receive Tasks on page 97.

MEP	Description	How to Model
Send request-response message to an external application	A client (service consumer) sends a request message to the web service (service supplier). The web service (service supplier) returns a response message to the client (service consumer). The web service may optionally return a fault message in the event of an error.	Service task (see Service Tasks on page 93).

Whether a message event performs a one-way operation or is part of a request-response operation is configured on the Properties view for the events.

Receiving a One-Way Request Operation

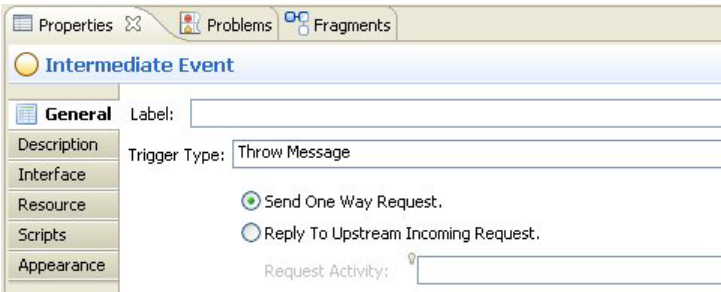
To configure an event to receive a one-way request operation, use either a start message event or a catch message event. The solution designer is responsible for configuring the event (for example, selecting a web service and mapping the input parameters to the process).



A receive task can be used in place of a catch message event (see [Receive Tasks on page 94](#)).

Sending a One-Way Request Operation

To configure an event to perform a one-way operation, use either a end message event or a throw message event. In the Properties view, ensure that **Send One Way Request** is selected:



- Throw message events are set to send one-way requests by default.
- A send task can be used in place of a throw message event.

Receiving a Request-Response Operation

There are several options for receiving a request-response operation. For example, a message start event can be paired with a message end event as follows:



In this example, the end message event is configured as follows:

The screenshot shows the 'End Event' configuration window. The 'General' tab is active. The 'Label' field contains 'Reply To: Start Message Event'. The 'Result Type' is set to 'Message'. Under the 'Request' section, the 'Reply To Upstream Incoming Request' radio button is selected. The 'Request Activity' dropdown is set to 'Start Message Event'. The 'Message Name' field is empty.

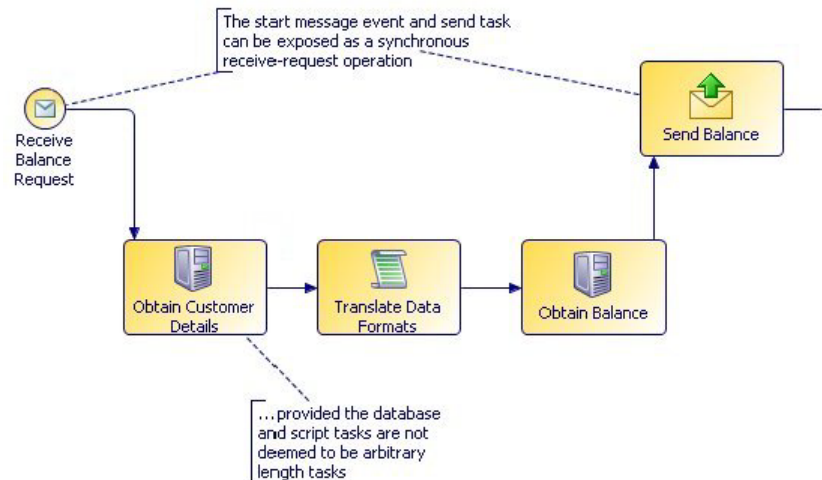


End message events are set to reply to upstream requests by default. In addition, if there is only one upstream request (for example, a start message event), the end event is automatically configured to reply to that request.

A send task can also be paired with a start message event as follows:



The tasks between the request and reply parts of the process should complete quickly enough that they do not cause the service calling the process to timeout. For this reason, avoid putting user tasks or manual tasks in between the request and reply parts of the process.



Chapter 8

Working with Process Data

This section describes how to use data that is used in processes such as data fields, parameters, and participants.

Topics

- [Creating Data Fields and Parameters, page 164](#)
- [Handling Incoming Messages Using Correlation Data, page 169](#)
- [Creating Declared Types, page 172](#)
- [Associating Process Data with Events and Tasks, page 173](#)
- [Associating Correlation Data with an Event or Task, page 176](#)
- [Setting Event and Task Visibility \(Private and Public\), page 178](#)
- [Using Process Data to Create Tasks, page 179](#)
- [Creating Participants, page 180](#)
- [Exporting Participants to an Organization Model, page 185](#)
- [Using Participants to Create Tasks, page 187](#)
- [Deleting Data Fields, Parameters, Participants, and Type Declarations, page 188](#)

Creating Data Fields and Parameters

Data fields identify the inputs and outputs of an activity. For example, an activity called "Process Student Course Request" could require a form with the list of courses the student wants to take as input. The availability is checked and a form that lists the courses they are enrolled in is output.

Parameters are input to or output from a source external to the process. For example, parameters are passed from a process to a sub-process.

Parameters can only be created at the process level. Data fields can be created:

- at the package level, where they can be shared amongst processes defined in that package.
- at the process level, where they can only be used by the parent process.
- at the activity level, where they can only be used by the parent activity. (Activity-level data fields are only supported for certain activity types in certain destination environments. See the appropriate implementation guide for more information.).



You can change a process-level data field to a parameter by right-clicking it, and selecting **Convert Data Field to Parameter**. The default mode for converted parameters is **In/Out**, meaning that they can be specified as either input or output.

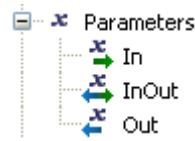
Data fields and parameters share the same basic types (Integer, Text, and so on), however they have different properties:

- Parameters can be specified as mandatory (they must be present at runtime).
- Both data fields and parameters can be specified as read only. For data fields, this means that they can only be assigned by their initial value setting. For formal parameters, it means that once input to the process they cannot be re-assigned (for example, in scripts or user tasks).
- Data fields and parameters can be an array of basic types (for example an array of Text).



In iProcess only, you will see a Warning message if an array field name exceeds 8 characters, and if a data field in general exceeds 15 characters.

- Parameters can also be specified as input, output or both by selecting the Mode (**In**, **Out**, or **In/Out**). The mode is indicated by the icon next to the parameter:



Adding Data Fields or Parameters to a Package or Process

There are two ways to add data fields or parameters to a package or process:

- in the Properties view table provided when the **Parameters** or **Data Fields** folder is selected in Project Explorer (this method is better if you need to create several data fields or parameters), or
- using the wizard to create one at a time.



See [Associating Participants with Activities on page 182](#) for information on how to highlight all the tasks in a process that have a particular data field or parameter assigned to them.

Using the Properties View

To create a new data field or parameter in the Properties view, do the following:

- In the Project Explorer, expand the package or process where you want to add a data field or parameter.

If the **Data Fields** folder is empty under a package, it will be hidden by default. This is because the preferred usage is to define Data Fields at the Process level.



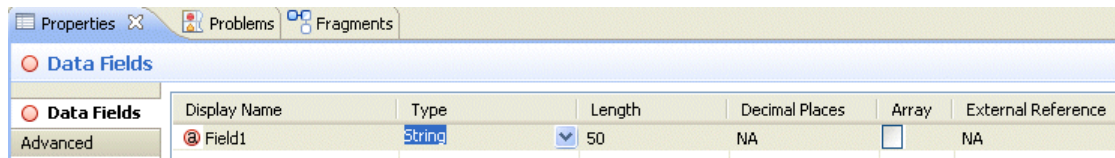
If you want to show the Data Fields folder under a package, regardless of whether it is empty or not, then you need to go to Project Explorer, and click the **View Menu** icon in the top right corner, then select **Customize View**. You can then deselect **BPM Package Data Fields** on the list of filters which are set.

- As appropriate, click **Data Fields** or **Parameters** folder in the Project Explorer. If you need to add a Data Field at the Package level, select **New > Data Field** as described in [Using the Wizards](#) below.

The Properties view is displayed.

- Click .

A new data field or parameter is added.



As appropriate, modify the properties.

Using the Wizards

To create a new data field or parameter, do the following:

- 1. In the Project Explorer, expand the package or process where you want to add a data field or parameter.
- 2. As appropriate, right-click **Data Fields** or **Parameters** and select **New > Data Field** or **New > Parameter**. The **New Data Field** or **New Parameter** dialog is displayed.



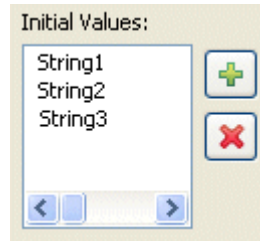
If you start this wizard from the **File > New** menu, the first dialog is the project and package dialog, where you must specify a valid project and package. This dialog is not displayed if you right-click at the process level to start the dialog; however you can click **Back** to display it.

- 3. Specify a label for the data field or parameter; you cannot specify leading or trailing spaces.
- 4. For parameters, specify the following:
 - **Mandatory** (*Parameter only*) Select this checkbox to specify that the parameter must be present when the process is started.
 - **Read Only** Select this checkbox to specify that the value of the data field or parameter cannot be modified after it is created.
 - **Mode** (*Parameter only*) Select whether the parameter will be an input (**In**) output (**Out**) or both (**In/Out**).



The parameters specified on the **Interface** tab for a user task are from the perspective of the *form*, not the *user*. This means that input parameters are sent to the form by the process, not the user. Output Parameters are sent to the form by the user.

5. Specify the type of data field or parameter you want to create:
 - **Basic Type** If you select this type, you can choose from **Text**, **Decimal Number**, **Integer Number**, **Boolean**, **Date**, **Time**, **Date Time** or **Performer**. For more information about using performer data types, see the appropriate implementation guide.
 - **Initial Value** For data fields, you can optionally specify an **Initial Value** by clicking in the provided text entry area and entering a value. For arrays, you can add more than one value:



You can add rows by clicking the  button. You can also delete rows by clicking the  button.

- **Allowed Values** For parameters, you can optionally specify the permitted input values (values that may be supplied by application starting an instance of the process). You can add rows by clicking the  button. You can also delete rows by clicking the  button.
 - **Declared Type** This option allows you to select from the declared types that you have already defined (see [Creating Declared Types on page 172](#)).
 - **External Reference** Allows you to refer to a business object defined in the Business Object Modeler.
6. Select whether you want the data field or parameter to be an array. Selecting the **Array** checkbox creates an array of the Basic Type that is selected. For example, if you select the **Array** checkbox and **Text**, you are defining the data field or parameter as an array of Text values.
 7. Click **Finish** to create the data field or parameter, or click **Next** to specify a documentation URL, description, or extended attributes.
 8. The **Documentation Url** and **Description** fields allow you to specify supplementary information about the data field or parameter that you have created.
 9. The **Extended** dialog allows you to specify extended attributes. For more information about extended attributes, see [Creating Extended Attributes on page 51](#).

The parameter that you created appears in the Project Explorer.

Adding Data Fields to an Activity

To create a new data field for an activity:

1. Select the activity for which you want to create a data field.
2. On the Properties view, click **Data Fields**.



Activities that do not support activity-level data fields (such as gateways) do not have a **Data Fields** tab on their Properties view.

A validation error is displayed if you define a data field for an activity on the **Data Fields** tab, but the destination environment selected for the parent process does not support activity-level data fields for that activity type.

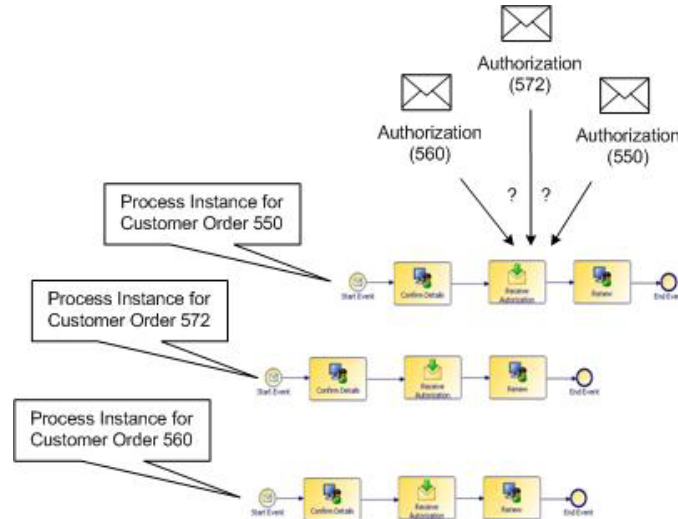
3. Click . A new data field is added.
4. Modify the data field's properties as appropriate.



You can only assign a value to an activity-level data field by using the **Scripts** tab on the activity. See the appropriate implementation guide for more information.

Handling Incoming Messages Using Correlation Data

In a process, a catch message event or a receive task waits for an incoming message to arrive. In the runtime environment there may be many instances of the process (each with different data), and many incoming messages (each with different data).



Correlation data must be used to ensure that each incoming message is received by the process instance to which it applies. The incoming data to a receive task or catch message event is compared to the correlation data in existing process instances to determine whether it applies to that process instance. You must initialize the correlation data with a value, either on the start message event, or subsequently (for example in a script task).

To illustrate how this works, consider the following example:



- The process begins with a start message event (the process is started upon receipt of a one-way message from an external web service). The business analyst decides whether elements within the incoming data from the web service can be used to uniquely identify instances of this process. If the incoming data can be used to identify the process (later on in the process when there are incoming messages), the business analyst creates a correlation data field (for example, **OrderRef**), and associates it with the start event on the **Interface** tab.

- After the user task, the receive task (**Receive Authorization**) waits for the correct incoming authorization. To do this, the business analyst associates the **OrderRef** correlation data field with the **Receive Authorization** task. When an incoming message is received, the incoming data is compared to the correlation data to make sure that it applies to that process instance.

After creating correlation data fields, and associating them with the relevant events or tasks, the correlation data must be mapped with the incoming data. This is the job of the solution designer. For more information, see the appropriate implementation guide.

Creating Correlation Data

There are several ways to create correlation data fields:

- using the Properties view table provided when the **Correlation Data** folder is selected in Project Explorer (this method is recommended if you need to create several data fields or parameters).
- using the wizard available either from the Project Explorer or from the **File** menu.



Additionally you can create correlation data fields by

- refactoring an existing data field into a correlation data field (right-click the data field and select **Convert to Correlation Data**).
- copying a data field and pasting it under the **Correlation Data** folder in the Project Explorer.

Creating Correlation Data Using the Properties View

To create a new correlation data field in the Properties view, do the following:

1. In the Project Explorer, expand the package or process where you want to add a correlation data field.
2. Click **Correlation Data** in the Project Explorer.

The Properties view is displayed.

3. Click

A new correlation data field is added.

Correlation Data					
Data Fields	Length	Decimal Places	Array	External Reference	Type Declaration
Advanced	50	NA	☐	NA	NA

As appropriate, modify the properties.



To prevent accidental reassignment of correlation data fields, they are set to read-only by default. If you want to assign a value to correlation data that does not come from an incoming message, deselect the **Read Only** checkbox.

See [Associating Correlation Data with an Event or Task on page 176](#) to assign correlation data to a start message event, catch message event, or receive task.

Creating Correlation Data Using the Wizard

To create a new correlation data field, do the following:

1. In the Project Explorer, expand the package or process where you want to add a data field or parameter.
2. Right-click **Correlation Data** and select **New > Correlation Data**. The **New Data Field** dialog is displayed.



If you start this wizard from the **File > New** menu, the first dialog is the project and package dialog, where you must specify a valid project and package. This dialog is not displayed if you right-click at the process level to start the dialog; however you can click **Back** to display it.

For information about how to complete the remainder of the fields in the wizard pages, see [Using the Wizards on page 166](#) (the properties of a correlation data field are exactly like those of a standard data field).

The correlation data that you created appears in the Project Explorer.

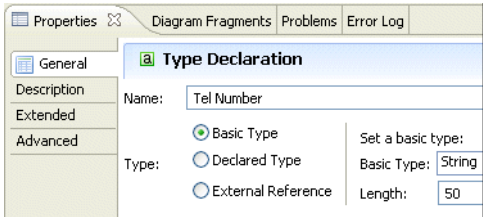


If a correlation data field has data that you want to continue to use in your process, but that you no longer want to use for correlation, you can convert the correlation data field to a "standard" data field, by right-clicking it and selecting **Convert Correlation Data to Data Field**.

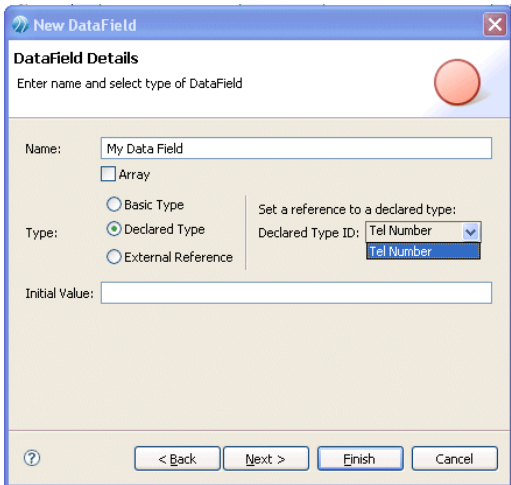
See [Associating Correlation Data with an Event or Task on page 176](#) to assign correlation data to a start message event, catch message event, or receive task.

Creating Declared Types

Declared types are used if you want to re-use a definition either when creating a data field or parameter. For example, you could create a declared type that is text that represents a telephone number:



This declared type is then available for use in defining data fields or parameters. For example:



Associating Process Data with Events and Tasks

There are many reasons to associate process data with tasks or events. For example:

- A data field needs to be displayed in a form.
- A process cannot be started unless a particular parameter is passed to a start event.

Drag and Drop

You can drag process data and drop it onto an event or task as follows:



You can drop process data onto a blank part of the process to create a new user task with the associated process data. For more information, see [Using Participants to Create Tasks on page 187](#).

1. In the Project Explorer, click the data field or parameter you want to associate with your activity, holding down the mouse button, drag the pointer to the event or task and release the mouse button.



You can select multiple data fields or parameters for drag and drop operations using the Ctrl (for single selection) or Shift (to select a range) keys.

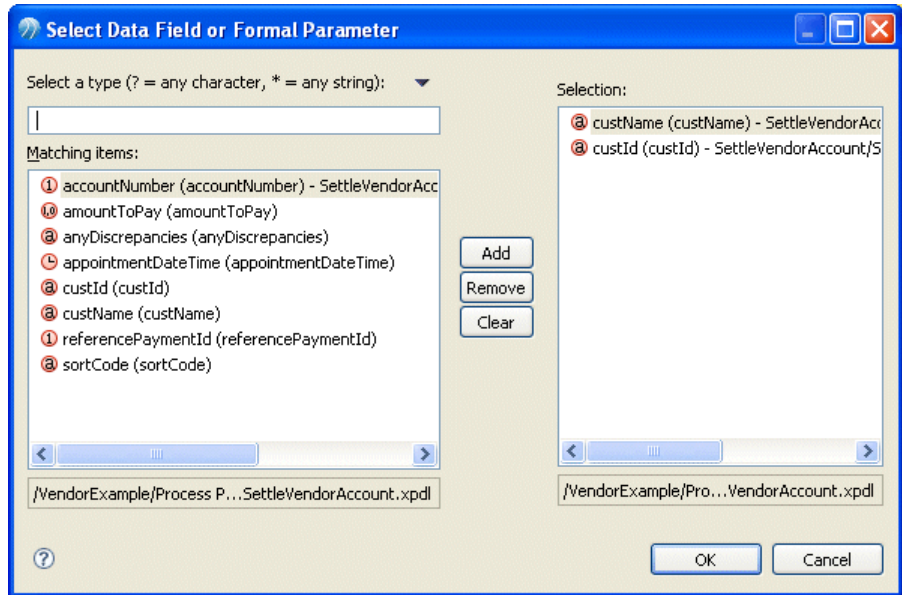
2. A menu is displayed with the following options:
 - **Add Data To View And Assign** Selecting this option adds the process data as an **In/Out** parameter on the **Interface** tab. This means that in the runtime environment, users will be able to display the associated form to view the field and also to assign new values to it.
 - **Add Data To View** Selecting this option adds the process data as an **In** parameter on the **Interface** tab. This means that in the runtime environment, users will be able display the associated form to view the field, however they can not assign new values to it.
 - **Add Data To Assign** Selecting this option adds the process data as an **Out** parameter on the **Interface** tab. This means that in the runtime environment, users will be able display the associated form to assign new values to the field.

Using the Interface Tab

Associating process data with events or tasks is done using the **Interface** tab.

1. Select the event or task.
2. In the Properties view, click the **Interface** tab.

3. Select whether you want the visibility of the event or task to be private or public - see [Setting Event and Task Visibility \(Private and Public\)](#) on page 178.
4. Click  to select the process data that you want to associate with the event or task. The **Select Data Field or Formal Parameter** dialog displays the list of available process data:



- By default all process data is available to a task. When you explicitly associate process data with an event or task, only the process data you associate with the event or task can be used by that task.
 - The process data displayed depends on what type of event or task is selected. Most events and tasks can have both data fields and formal parameters associated with them, however receive tasks and events of type None can have only formal parameters associated with them. End events of type Message can only have formal parameters if the selected WSDL has both input and output parameters.
5. The process data you select is added to the table of data. Select whether you want the data to be mandatory.



The mandatory setting on the Interface tab for a formal parameter overrides the mandatory setting in the Properties view for the formal parameter. This allows complete freedom in designing the process - you can define a formal parameter as mandatory in one place in a process, and optional in another.

Use the space provided if you want to add an optional usage description of the process data. Selected parameters also display their mode (In, Out, or In/Out). You can change the mode by selecting from the drop-down list.

Associating Correlation Data with an Event or Task

Catch message events and receive tasks must have correlation data associated with them. Start message events can optionally have correlation data.

This is done on the **Interface** tab as with other process data. There is a special section on the **Interface** tab for correlation data:

▼ Correlation Data

☐ No correlation data initialization required.

Correlation Data Name	Mode	Description
[All Correlation Data]		



If you do not want a start message event to have correlation data, select the **No correlation data initialization required** check box in the Correlation Data section of the **Interface** tab. If you select this check box, you must ensure that correlation data is initialized (for example, using a script) before it is required in the process.

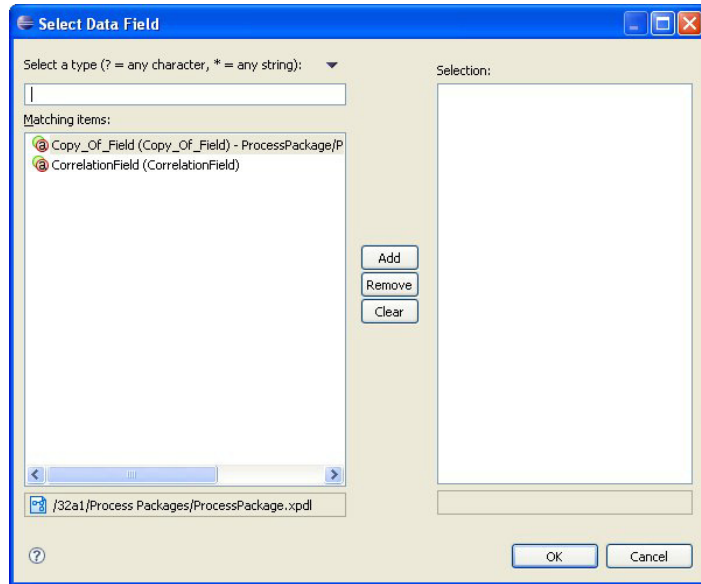
By default, all correlation data is associated with an event or task, but you can explicitly associate only selected correlation data as described in this section.

1. In the Process Editor, select the event or task to which you want to add correlation data.
2. Click the **Interface** tab.



If the **Interface** tab does not display the **Correlation Data** section, make sure that the event or task you have selected is one of the types listed previously. Other types of event or task do not require correlation data and therefore do not display this section on the **Interface** tab.

- Click  to select the correlation data that you want to associate with the event or task. The **Select Data Field** dialog displays the list of available correlation data fields.



- When you have finished selecting correlation data, click **OK**.
- Select the mode of the correlation data.



Depending on the type of event or task, TIBCO Business Studio provides appropriate defaults. For example, start message events are usually where correlation data is initialized, so the default mode for start message events is **Initialize**.

- **Correlate** (*Intermediate Catch Message Event and Receive Task Only*) The specified incoming data is compared to that of existing process instances to ascertain the target of the request.
- **Initialize** Specifies that the correlation data is expected to receive its value from the incoming data. For a start message event, a process instance is started, then the correlation data field is initialized with a value from the incoming data.
- **Join** (*Start Event or Received Tasks without incoming flow Only*) Causes the correlation data (mapped from the input data) to be compared to the correlation data for any existing process instances. If no matching process instances are found, a new process instance is started. If the correlation data matches that of an existing process instance, the request is directed to that instance (a new process instance is not started).

Setting Event and Task Visibility (Private and Public)

The visibility of an event or task (whether it is private or public) controls whether process information (such as required parameters) is available to an external process or application. Setting the visibility to public results in a subset of events or steps that are then available to external processes or applications.



The runtime effect of this setting is entirely destination specific.

For each event or task, you can specify a list of expected input fields (on the **Interface** tab), and define the URL of the document describing the purpose of the public event or task (on the **Description** tab).

By default, events and tasks in a process are private; in a process interface, they are public. If you want to change the visibility of an event or task, do the following:

1. Select the event or task.
2. On the Interface tab, select the visibility (**Private** or **Public**).
3. Click  to add parameters to the event or task. Use the Mandatory checkbox to specify parameters that must be present.
4. *(Optional)* Select the **Description** tab. Enter a URL that provides documentation describing how the event or task is used.

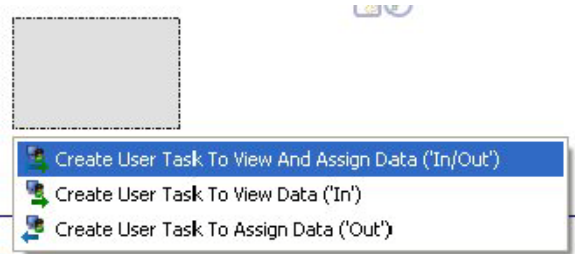
Using Process Data to Create Tasks

If you have created data fields or parameters, you can drag them to a blank area of a process to automatically create a user task:

1. In the Project Explorer, click the data field or parameter, and holding down the mouse button drag the pointer to a blank part of the process and release the mouse button:



You can select multiple data fields or parameters for drag and drop operations using the **Ctrl** (for single selection) or **Shift** (to select a range) keys.



2. A menu is displayed with three options:
 - **Create User Task To View And Assign Data** Selecting this option creates a user task with the process data as an **In/Out** parameter on the **Interface** tab. This means that in the runtime environment, users will be able to display the associated form to view the field and also to assign new values to it.
 - **Create User Task To View Data** Selecting this option creates a user task with the process data as an **In** parameter on the **Interface** tab. This means that in the runtime environment, users will be able display the associated form to view the field, however they can not assign new values to it.
 - **Create User Task To Assign Data** Selecting this option creates a user task with the process data as an **Out** parameter on the **Interface** tab. This means that in the runtime environment, users will be able display the associated form to assign new values to the field.

Creating Participants

Participants are used to identify who or what performs an activity. For example, in a hiring process, a person (Human Participant) interviews the candidate and an email system (system participant) sends out an automatic follow-up reminder. There are the following options for creating participants:

- Use the basic types of participant that are provided by TIBCO Business Studio.
- Create your own organization model, and use the positions from the organization model for your participants.

You can also refactor the participants in your process into an organization model (see [Exporting Participants to an Organization Model on page 185](#)).

Basic Types

There are several basic types of participant:

- **Role** - identifies the role responsible for performing an activity. For example, in a financial institution there may be roles such as Reconciler, Manager, and so on.
- **Organizational Unit** - identifies the department or unit within an organization that performs an activity. For example, Legal, Marketing and so on.
- **Human** - identifies a specific person or user that performs an activity.
- **System** - identifies an activity that is performed by a system.
- **Organization Model Query** - allows you to enter a query using a script or expression. This is evaluated when a referencing task is executed at run-time, so the actual participant is resolved and the activity dispatched and offered to the participant. A query could resolve to a participant in the package/process or to an entity in the organizational model.

Participants from the Organization Model

Besides the basic types of participant described in the previous section, you can also create participants by creating an external reference to types defined in a model of your own organization.

Type: ☐ Basic Type
☒ External Reference

Set a reference to a type defined in the domain model:
In external Organization Model ...

You can create an external reference to the following parts of the organization model:

- Positions
- Groups
- Organization Units
- Types from the default meta model

You can also create an organization model from the participants you have defined in your process as described in [Exporting Participants to an Organization Model on page 185](#).

Creating a Participant

To create a new participant, do the following:

1. In the Project Explorer, expand the package where you want to add a participant.
2. Right-click **Participants** and select **New > Participant**. The **New Participant** dialog is displayed.
3. Click the **Back** button if you need to change either the name of the **Project** and **Package** where the participant will be created. If you want to change either, click the **Project** or **Package** button.

Participants can be created at either the package level or at the process level. Creating them at the package level is recommended as it enables them to be shared amongst processes. Select the **Process** checkbox and specify a process if you want to create the participant at the process level.



If the Participants folder is empty at the Process level, it will be hidden by default. This is because the preferred usage is to define Participants at the Package level.

4. Click **Next**.
5. Specify the **Name** and **Type** of the participant (either a basic type, or an external reference as described previously in this section) and click **Finish**.
 - To create a basic type, select **Basic Type** and choose from **Role**, **Organization_Unit**, **Human**, **System** and **Organization Model Query**, and click **Finish**.

If you select the **Organization Model Query** button, you can then enter the Organization Model Query Script in Resource Query Language (RQL) using a script or expression in the General tab of the Properties view. This is evaluated when a referencing task is executed at run-time, so the actual

participant is resolved and the activity dispatched and offered to the participant. A query could resolve to a participant in the package/process or to an entity in the organizational model.

- To create an external reference to an organization model, select **External Reference**, and click  to select a type from the organization model.

Choose a type from those shown in Matching Items, or key in the first few characters of the name you are looking for in the field under **Select type(s)(? = any character, * = any string** and choose from those shown. Click **OK**.

The participant that you created appears in the Project Explorer.

Associating Participants with Activities

You can associate a participant with an activity to identify who or what performs the activity. You can do this either by dragging and dropping the participant onto the activity or in the Properties view for the activity.

Drag and Drop You can drag a participant onto an activity as follows:

1. Expand participants in the Project Explorer.
2. Click the participant you want to associate with your activity, holding down the mouse button, drag the pointer to the activity and release the mouse button.



- You can select multiple participants for drag and drop operations using the **Ctrl** (for single selection) or **Shift** (to select a range) keys.
- You can also select as a participant a data field or formal parameter of the type **Performer**.

3. A menu is displayed with two options:
 - **Add Task Participant(s)** Selecting this option adds the participants to any existing participants for the activity.
 - **Set Task Participant(s)** Selecting this option clears any existing participants associated with the activity, before setting the participants to those you selected.

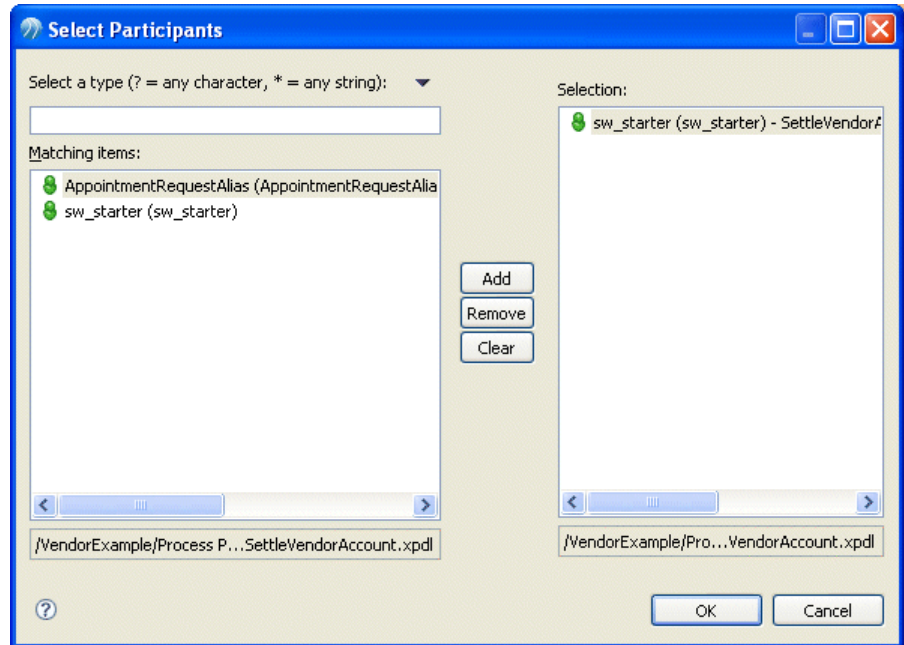
Properties View To associate a participant with an activity do the following:

1. In the Process Editor, highlight the desired activity.

2. Either:

- In the Properties view, click .
- Right-click the activity, and select **Participant**.

The Select Participants dialog is displayed:



3. Highlight participants you want to select and click **Add** to move them to the **Selection** column. When you have finished selecting Participants, click **OK**.



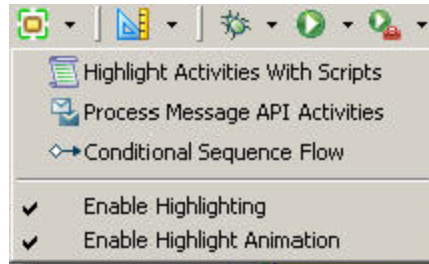
- You can select multiple participants by pressing either the **Ctrl** (for single selection) or **Shift** (to select a range) keys while making your selection.
- The wildcard **?** returns all matching participants. Use the ***** wildcard to restrict the results (for example, ***2** to return all Participants ending in 2. Note that the wildcard ***** by itself does not return any results; it only works in conjunction with a string.
- You can also select as a participant a data field of the type **Performer**.

4. The participants you selected are displayed in the Properties view and also when you hover the pointer over the activity in the Process Editor.



To see what tasks a participant is assigned to, click in the Process Editor for the process. On the toolbar, you will see a button: 

Click on this to see the dropdown, and select **Enable Highlighting**.



Alternatively, you can select **Diagram > Highlighting > Enable Highlighting**.

Select the participant in Process Explorer, and in the Process Editor the tasks which use this participant will be highlighted (framed with green lines). Other tasks in the process will be grayed out.

This behavior also applies to data fields and parameters.



Highlights are cumulative.

- If you select more than one participant, data field or parameter in Project Explorer, only tasks which reference all of those selected will be highlighted.
- If you select "Activities With Scripts" followed by "Process Message API Activities" then the diagram objects highlighted will be process message API activities that have scripts.

The current highlight filter is shown when you use the Process Editor mouse-over tooltip.

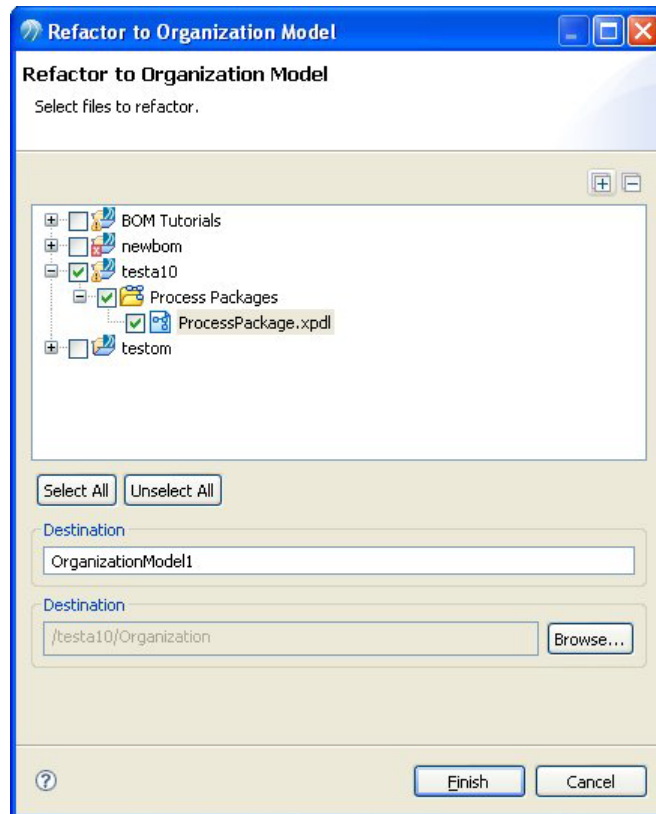
Highlights are removed when anything other than the highlighted diagram objects are selected.

Exporting Participants to an Organization Model

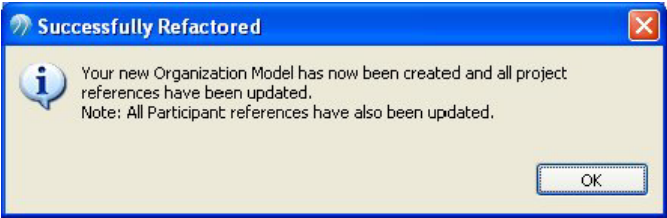
To create an organization model based on the participants in a process package, or a number of process packages, use the procedure described in this section.



- Process packages with errors cannot be selected as the basis for organization model creation.
 - Processes must contain only one pool, but can have many lanes.
 - If a process package references another process package, the referenced process package is automatically selected.
1. Right-click the XPDL package file in the Project Explorer and select **Refactor > To Organization Model**.
 2. In the resulting dialog, select the process packages that you want to refactor into an organization model, and click **Finish**:



3. The following message indicates that the process packages were correctly refactored:



The existing participants are changed from basic types to external references to the organization model.

Participant to Organization Model Mapping

The following table shows how participants that *are not* referenced in a process appear in the exported organization model.

Process Participant Type	Organization Model Object
Human	Position in default organization model unit
Organization unit	Top-level organization unit
Role	Top-level group
System	Ignored

The following table shows how participants that *are* referenced in the process (for example, in a user task) appear in the exported organization model.

Process Participant Type	Organization Model Object
Human participant	Position in organization unit (unit is named after the lane in the original process)
Organization unit	Organization unit within an organization unit (the containing unit is named after the lane in the original process)
Role	Top-level group
System	Ignored

Using Participants to Create Tasks

If you have participants, you can drag them to a blank area of a process to automatically create several types of task:

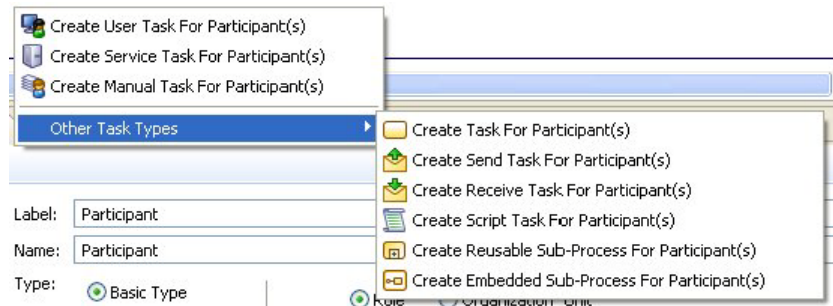
1. In the Project Explorer, click the participant, and holding down the mouse button drag the pointer to a blank part of the process and release the mouse button.



You can select multiple participants for drag and drop operations using the **Ctrl** (for single selection) or **Shift** (to select a range) keys.



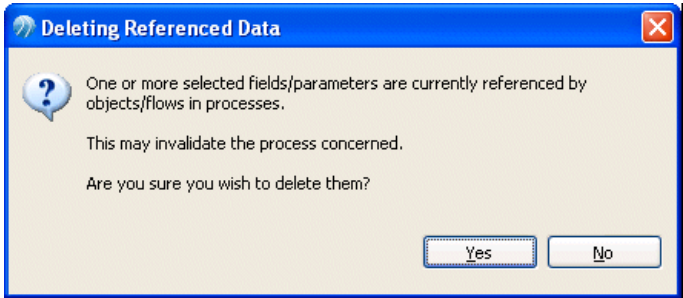
2. A menu is displayed with the following options:



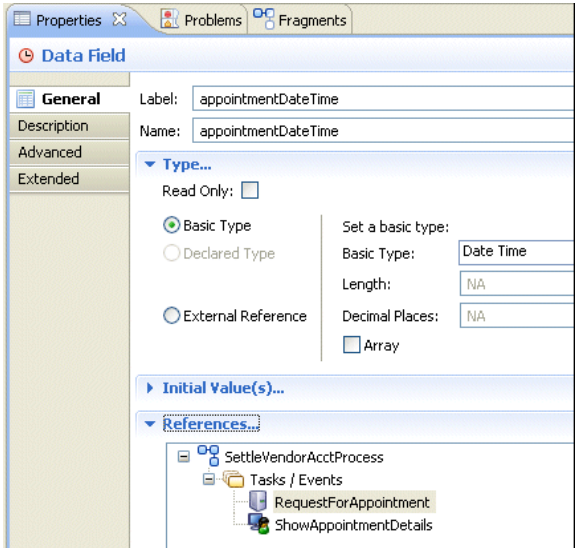
- **Create User Task For Participant(s)** Selecting this option creates a user task with the selected participants.
- **Create Service Task For Participant(s)** Selecting this option creates a service task with the selected participants.
- **Create Manual Task For Participant(s)** Selecting this option creates a manual task with the selected participants.
- **Other Task Types** This submenu contains more task types that you can create from a participant.

Deleting Data Fields, Parameters, Participants, and Type Declarations

You can delete a project object such as a data field, parameter, participant, or type declaration by right-clicking it in the Project Explorer and selecting **Delete**. However, if the project object has been associated with another object such as an activity or a sequence flow, the following message is displayed:



If you click **Yes**, the project object (in this example, a data field) is deleted, but it will still be referenced in XPDL for the project, which can cause problems. If you click **No**, the project object is not deleted. You can then go to the Properties view and examine the references to that object:



In this example, the data field is referenced in the service task **RequestForAppointment** and in the user task **ShowAppointmentDetails**. Click **Go To** to show that task in the Process Editor.

Chapter 9

Using Sub-Processes and Process Interfaces

This chapter describes how to work with sub-processes and process interfaces in TIBCO Business Studio.

Topics

- [Reusable Sub-Processes, page 190](#)
- [Embedded Sub-Processes, page 194](#)
- [Refactoring Sub-Processes, page 197](#)
- [Inline Sub-Processes \(Package Optimization\), page 208](#)
- [Creating an Inline Sub-Process, page 209](#)
- [Inline Sub-Process in Detail, page 211](#)
- [Criteria for Inline Sub-Processes, page 215](#)
- [Process Interfaces, page 218](#)
- [Creating a Process Interface, page 221](#)
- [Making a Sub-Process Call to a Process Interface, page 228](#)
- [Modifying a Process Interface, page 231](#)
- [Using a Dynamic Sub-Process, page 232](#)

Reusable Sub-Processes

When identifying the process components you require for your business process, it is important to analyze the entire business. One way to do this is by thinking in terms of the products, services and resources (by considering the specifications for each).

To facilitate the re-use of process components, an activity (or several activities) can call another process as a reusable sub-process. The reusable sub-process could be a process that you have already created, or you can refactor activities in your current process into a reusable sub-process (see [Refactoring Sub-Processes on page 197](#)).

There are three different ways of creating a call to a reusable sub-process:

- By refactoring objects in your process (see [Refactoring Sub-Processes on page 197](#)).
- By dragging a process from the Project Explorer and dropping it onto your process (described in this section).
- Using the reusable sub-process tool from the palette (described in this section).



If you modify a sub-process (for example, by adding a parameter) and that sub-process is referenced by a main process in a different package, validation in the main process does not occur until you save the sub-process package.

Drag and Drop

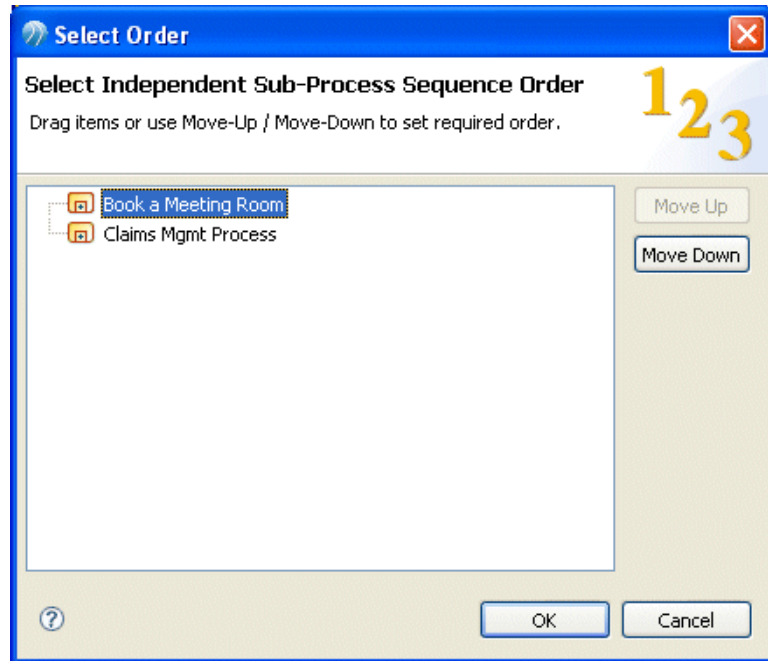
To create a reusable sub-process call using drag and drop, do the following:

1. Expand the Project Explorer to locate the process that you want to be the sub-process.
2. Click the intended sub-process, holding down the mouse button, drag the pointer to the calling process (open in the Process Editor), and release the mouse button.



You can select multiple processes for drag and drop operations using the Ctrl key.

3. If you are dropping more than one process, a menu is displayed with two options:
 - **Create Sub-Process Task Sequence** Selecting this option allows you to create sub-process tasks connected by sequence flow. The following dialog is displayed to allow you to control the order of the tasks:



Use the **Move Up** and **Move Down** buttons or drag and drop to control the order of the tasks. When you are finished, click **OK** to place the tasks.

- **Create Unsequenced Sub-Process Tasks** Selecting this option places the tasks in the process without a connecting sequence flow.

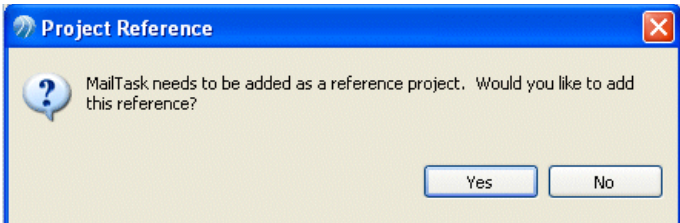
Palette To create a reusable sub-process call using the palette, do the following:

1. In the Process Editor, select the **reusable sub-process** tool.
2. Click in the process where you want to place the activity that calls the sub-process.

- 3. On the Properties view for the activity, browse for the process you want to call as a sub-process.

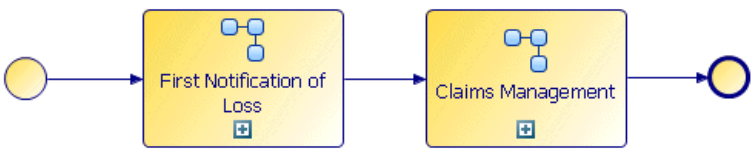


If the process you select is not in the current project, you are prompted to create a project reference:

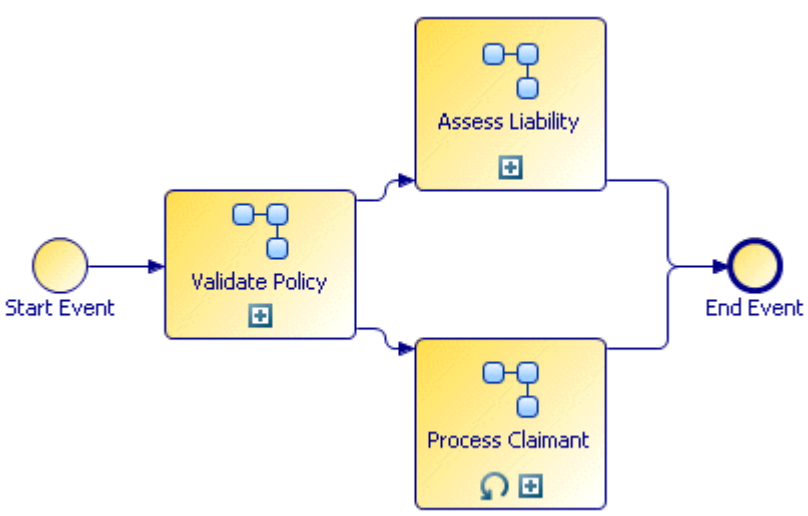


Example

The following example demonstrates how a top-level process could use sub-processes in an insurance environment:



Each of the activities in the process calls sub-processes that are executed in the runtime environment. For example, the sub-process called from the Claims Management activity could look like this:



Note the following about this sub-process:

- The Validate Policy sub-process can be re-used. For example, when a customer calls to renew their policy, the first step of the Renewal Process could be to confirm that they have an existing policy. The Validate Policy Activity could be re-used for this purpose in the Renewal process.
- The Process Claimant activity has a **Loop** Activity Marker, that indicates that the activity is repeated for each Claimant.
- The *actual* sub-process called by the Assess Liability activity is a manual Process. However, the *to be* sub-process (planned for the future), could be implemented as an automatic process that consists of a series of questions used to determine liability. Simulation could be used to demonstrate the performance and cost-savings of migrating the Assess Liability activity to an automatic process.

Expanding a Sub-Process

An activity that calls a sub-process looks like this:



To expand the sub-process associated with this activity, either click the  symbol, or do the following:

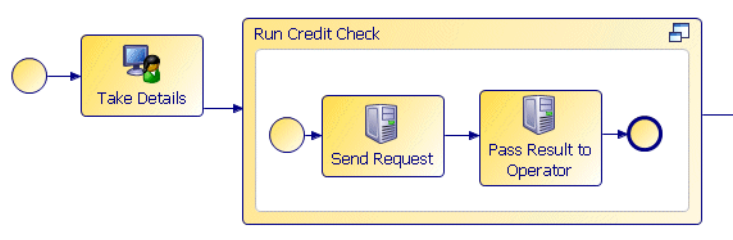
1. In the Process Editor, select the activity that calls the sub-process.
2. In the Properties view, select the **General** tab.
3. Click **Open Sub-Process**:

Invocation of Sub-Process:	(Open Sub-Process)
Sub-Process location:	MyProcessPackage1.xpdl
Sub-Process name:	My Process...

4. The sub-process opens in the Process Editor.

Embedded Sub-Processes

An embedded sub-process is one that is fully contained within the parent process; it does not exist as a separate process:



In this example, Run Credit Check is an embedded sub-process. This implies that running a credit check is an activity that is not needed by other processes. If you subsequently decide that you want to be able to run a credit check from within other processes, you can expose the embedded sub-process as a reusable sub-process by refactoring it. See [Refactoring Sub-Processes on page 197](#).

To create an embedded sub-process, refactor one or more objects in your process as described in [Refactoring Sub-Processes on page 197](#).



If you want to use the chained execution resource pattern, you can do so by selecting the **Chained Execution** check box in the Properties view for the embedded sub-process. For more information, see [Chained Execution on page 116](#).

Local Data Fields

You can add data fields that are local to a sub-process (they are not used in the process that contains the embedded sub-process). By default, activities in the embedded sub-process have access to all local data and process data. However, using the **Interface** tab, a subset of data can be selected.



Embedded sub-process local data fields are not allowed when deploying to iProcess destinations.

To add local data fields to an embedded sub-process:

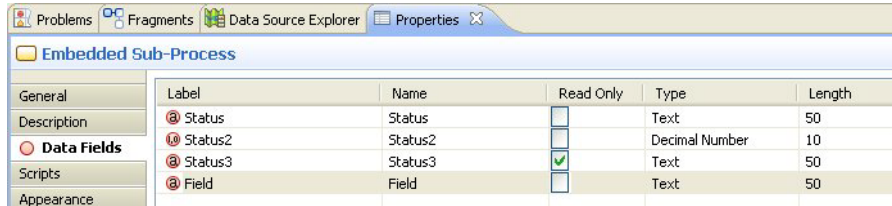
1. Click the **Data Fields** tab in the Properties view for the embedded sub-process.

- At the right of the **Data Fields** tab, click  to add local data fields. The properties of the data fields that you create are the same as for process data (see [Creating Data Fields and Parameters on page 164](#)).



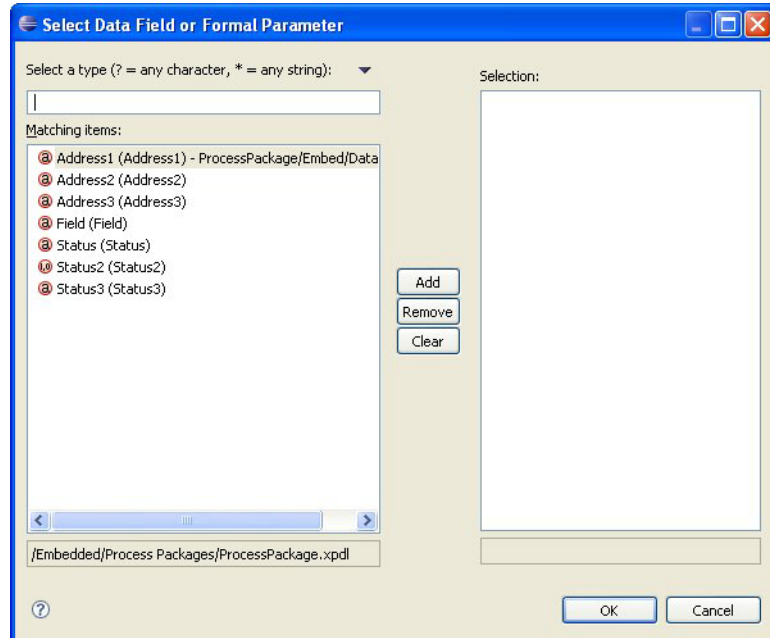
Local data fields are not displayed in the Project Explorer.

For example:



Label	Name	Read Only	Type	Length
Status	Status	☐	Text	50
Status2	Status2	☐	Decimal Number	10
Status3	Status3	☒	Text	50
Field	Field	☐	Text	50

In this example, the data fields **Status**, **Status2**, **Status3**, and **Field** are local to the embedded sub-process. On the **Interface** tab for activities within the embedded sub-process, you can access the local data as well as the process data:



In this case the address data fields (process data) and the local data defined on the **Data Fields** tab are both available.



You can quickly create an embedded sub-process that has local data by using the fragment **BPMN Process Fragments > Basic Fragments > Embedded Sub-Process with Data Fields**. For more information about fragments, see [Using Process Fragments on page 69](#).

Using Local Data Fields in Loops

When using local data fields in loops, the value taken by the local data field during each iteration depends on the type of loop.

- Multi-instance loop Each instance of a multi-instance loop has a separate instance of the local data field, allowing parallel loops to execute without interfering with each other. However, the local data fields cannot be used in the loop expression.
- Standard loop The local data field is shared between all instances of a standard loop. It can be used to pass data between instances, and can be used in the loop conditional script.



You cannot initialize a local data field prior to the first "Before" condition test.



For both multi-instance and standard loops, the task initiate script runs just before executing the task for each iteration of the loop.

Refactoring Sub-Processes

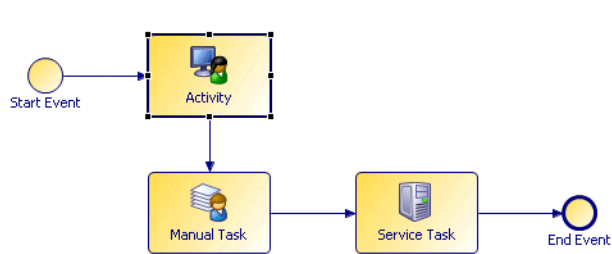
Refactoring a sub-process allows you to do the following:

- Create a new embedded sub-process from selected objects.
- Create a new reusable sub-process from selected objects and replace the selected objects with a call to the newly created sub-process.
- Convert an existing embedded sub-process (and its contents) into a reusable sub-process and replace it with a call to the newly created sub-process.

Creating a New Embedded Sub-Process

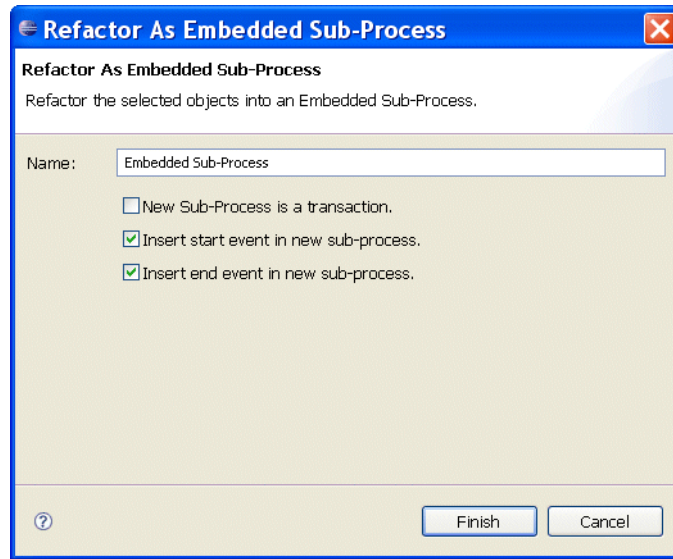
To create a new embedded sub-process from objects in your process, do the following:

1. Select the objects that you want to put in the embedded sub-process. For example:



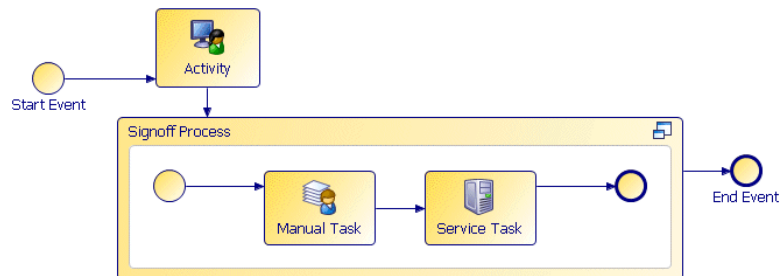
2. Right-click and select **Refactor > Move into New Embedded Sub-Process**.

3. The following dialog is displayed:



Complete the dialog as follows:

- Enter a name for the embedded sub-process that you want to create.
 - Select the **New Sub-Process is a transaction** checkbox if you want the new sub-process to become a transaction and therefore be under transaction control (see [Transactions on page 101](#)).
 - Select the **Insert start event in new sub-process** and **Insert end event in new sub-process** checkboxes to control whether start and end events are added to the refactored sub-process.
4. Click **Finish**. The objects that you selected are placed within a new embedded sub-process (with start and end events if those options were selected):



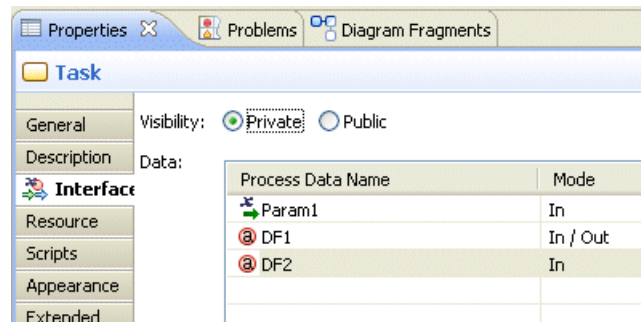
Creating a New Reusable Sub-Process

To create a new reusable sub-process from objects in your Process, do the following:

1. Select the objects that you want to put in the reusable sub-process. For example:



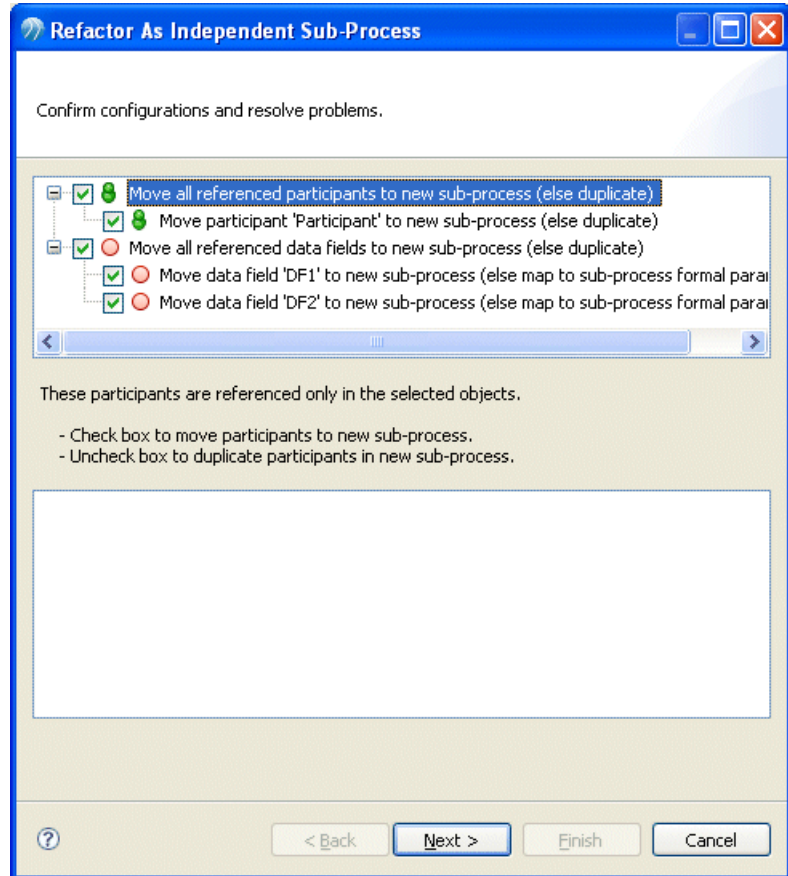
In this example, the two user tasks have the following parameters (two data fields and one formal parameter):



The user tasks also have a participant associated with them.

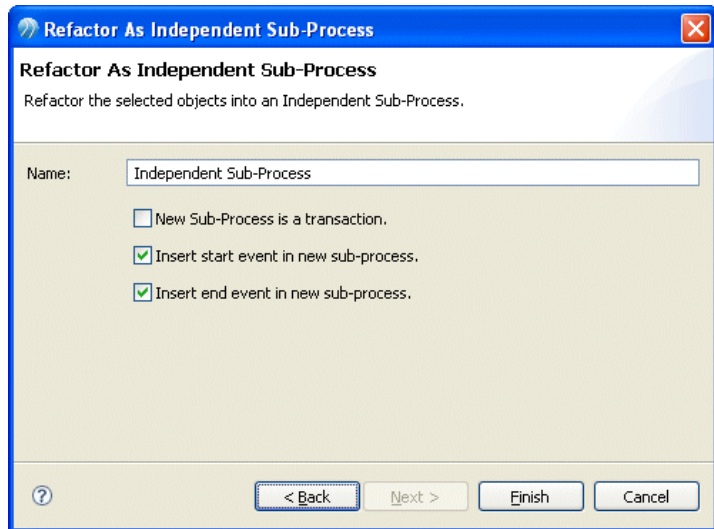
2. Right-click and select **Refactor > Extract into New Reusable Sub-Process**.

- Because the data fields and participants are only referenced in this activity (and not used by any other activities), the following message is displayed:



- For the participant, TIBCO Business Studio gives you the option of either moving this participant into the reusable sub-process, or duplicating the participant in both the parent process and the sub-process.
- If you select the data fields, they are moved into the Sub-Process.
- If you do not select the data fields, they are created as formal parameters in the sub-process (and mapped to those formal parameters).

4. The following dialog is displayed:

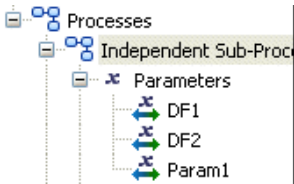


Complete the dialog as follows:

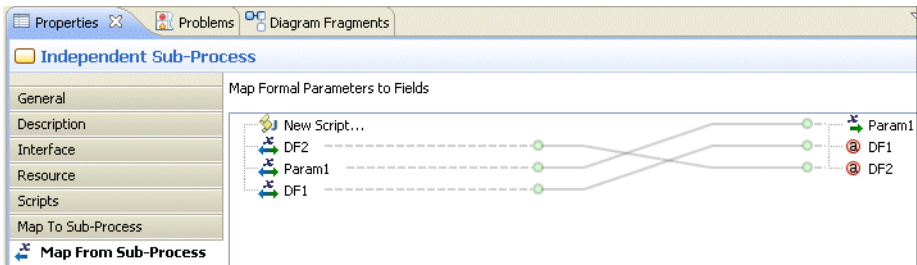
- Enter a name for the reusable sub-process that you want to create.
 - Select the **New Sub-Process is a transaction** checkbox if you want the new sub-process to become a transaction and therefore be under transaction control (see [Transactions on page 101](#)).
 - Select the **Insert start event in new sub-process** and **Insert end event in new sub-process** checkboxes to control whether start and end events are added to the refactored sub-process.
5. Click **Finish**. The objects that you selected are copied to the new sub-process and the selected objects are replaced with a task that calls the reusable sub-process.



If you chose to create formal parameters for the data fields, you can see them in the Project Explorer, under the sub-process:



In addition, if you click task in the parent process that calls the sub-process and go to the Properties view, you can see the mappings that have been created between data fields and formal parameters of the parent process and the formal parameters created in the sub-process. For example:



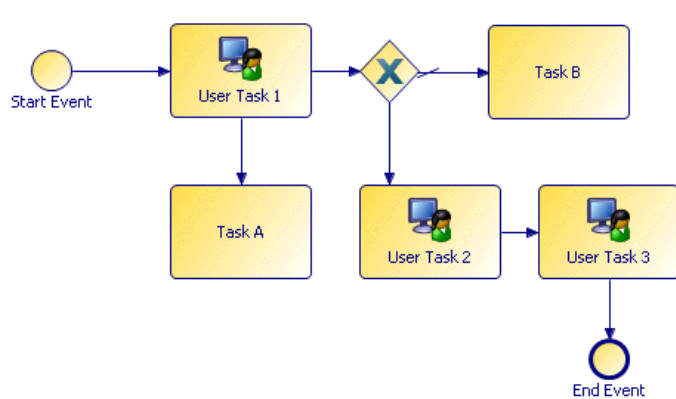
Notes on Refactoring Objects into Reusable Sub-Processes

Note the following when refactoring objects into reusable sub-processes:

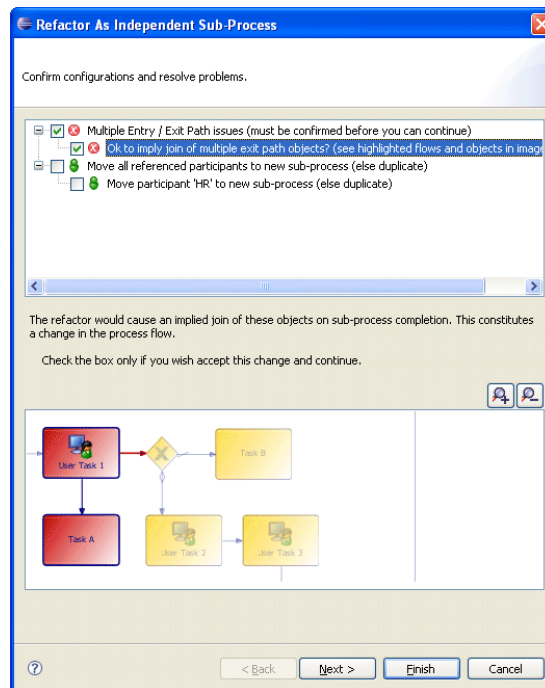
- Formal parameters in the parent process are created as formal parameters in the sub-process.
- If a data field is referenced only in the selection that you are refactoring, you have the option of moving or copying it. Data fields that are *copied* into the sub-process are created as formal parameters. Data fields that are *moved* into the sub-process are created as data fields.
- A mapping is created between data fields and formal parameters of the sub-process and any corresponding formal parameters that are created in the sub-process. You can view this mapping by selecting the task that calls the sub-process and clicking the **Map To Sub-Process** and **Map From Sub-Process** tabs in the Properties view.

Changes in Process Logic

When refactoring objects into a reusable sub-process, you can potentially change the logic of the process flow. Consider the following process:



In this process, flow proceeds from **User Task 1** to the gateway without necessarily waiting for **Task A** to finish. If **User Task 1** and **Task A** are refactored into a reusable sub-process, an end event is inserted into the sub-process, effectively synchronizing the flow. TIBCO Business Studio prompts you to confirm this change to the process:



Inline Sub-Process Content

This option creates an embedded sub-process from a reusable sub-process. The reason for making a reusable sub-process inline is that in some cases the tasks in the sub-process execute relatively quickly and the overhead of invoking a sub-process can be comparatively high.

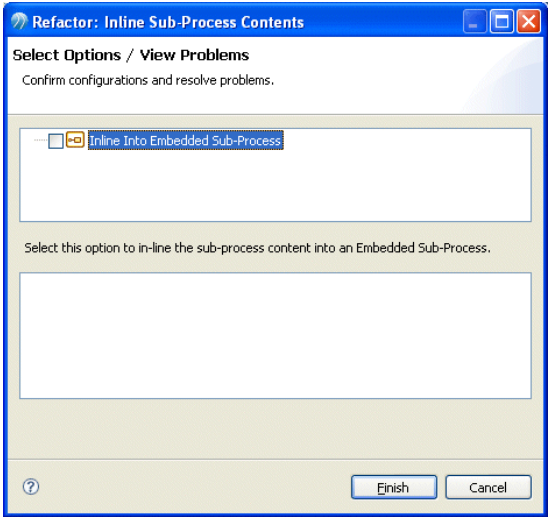
To refactor a reusable sub-process into an embedded sub-process, do the following:

1. Right-click the reusable sub-process task, and select **Refactor > Inline Sub-Process Content**.



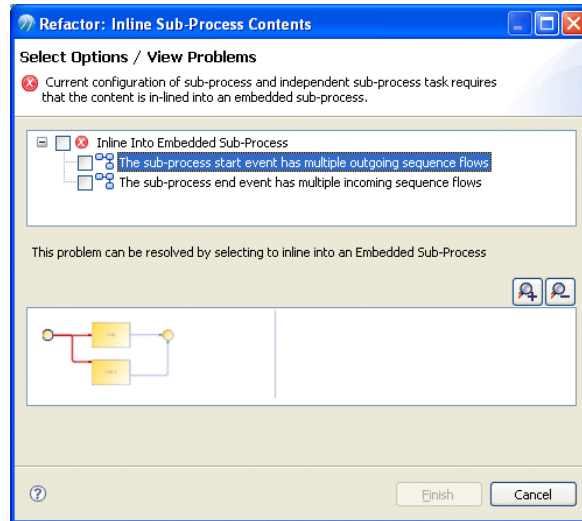
Only a single level of a process hierarchy can be made inline at a time. If the sub-process you want to make inline contains a sub-process task, after refactoring, it will be brought into the embedded sub-process as a sub-process task. (the refactor does not extend into additional levels of the process hierarchy).

2. The reusable sub-process called by the selected task is analyzed, and a dialog is displayed with the results:
 - If there are no problems with the refactoring operation, the following dialog is displayed:



Select **Inline into Embedded Sub-Process** and click **Finish**.

- If there are problems with the refactoring, but the problems can be resolved by making the sub-process inline, a dialog is displayed showing the problems:

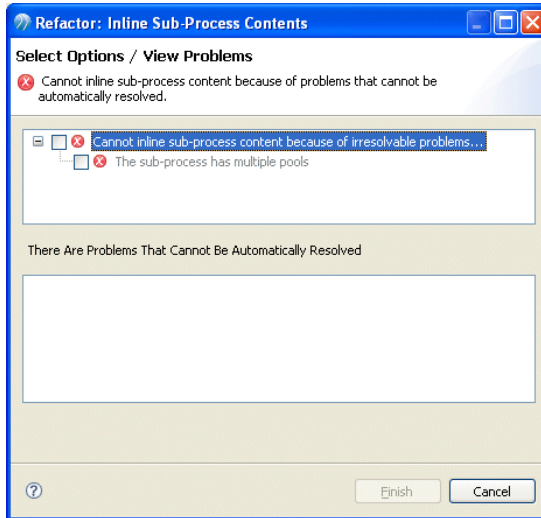


The possible problems displayed in this dialog are similar to those for process package optimization (see [Problems that Prevent Optimization on page 215](#)), except that some of these problems that are not resolvable for package optimization are resolvable by refactoring into an embedded sub-process (such as those reported in the previous dialog).

Also, problems dealing with nesting of sub-processes are not applicable because manual refactoring affects only a single level of sub-process hierarchy at a time.

You can see the location of each problem by highlighting it. If you select all of the problems in this dialog, the **Finish** button is enabled and you can refactor the sub-process.

- If there are problems that cannot be resolved by refactoring, a dialog similar to the following is displayed:



Because the errors cannot be resolved, click **Cancel**, resolve the problems manually, and retry the refactoring.

3. If the sub-process was able to be refactored, its contents are placed in an embedded sub-process, however the sub-process that was made inline is not removed.



The following are potential consequences of a refactoring. For more information, see [Inline Sub-Process in Detail on page 211](#):

- References to sub-process parameters in the sub-process content are swapped for the calling process data fields that are mapped to them.
- Sub-process data fields and unmapped parameters are copied up to the calling process and renamed with a sequence number if the data field or parameter already exists in the calling process.
- Sub-process participants are copied to the calling process (if they do not already exist).
- Type declarations referenced by data copied up from the sub-process are copied to the package of the calling process *only* if the sub-process is in a different package.
- If the sub-process implements a process interface (see [Process Interfaces on page 218](#)), the start or intermediate events that implement interface-defined events are changed so that they are no longer flagged as such.

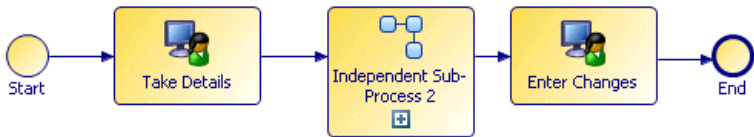
Inline Sub-Processes (Package Optimization)

Creating an inline sub-process from a reusable sub-process is a way of optimizing the process. At runtime, the objects such as activities and events contained in the reusable sub-process are brought into the top-level process and executed there, rather than by making a call to the sub-process.



A reusable sub-process can also be refactored into an embedded sub-process (see [Inline Sub-Process Content on page 204](#)).

The reason for inline sub-processes is that in some cases the tasks in the sub-process execute relatively quickly and the overhead of invoking a sub-process can be comparatively high. For example, consider the following process:



The reusable sub-process called by the second task consists of a single service task:



If the call to the reusable sub-process is made inline and the package optimized, the process at runtime would execute as follows:

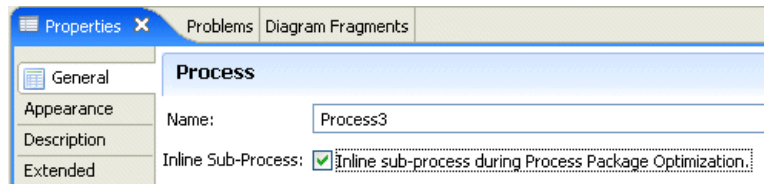


This is functionally equivalent to the original process, except that it does not have the sub-process invocation.

Creating an Inline Sub-Process

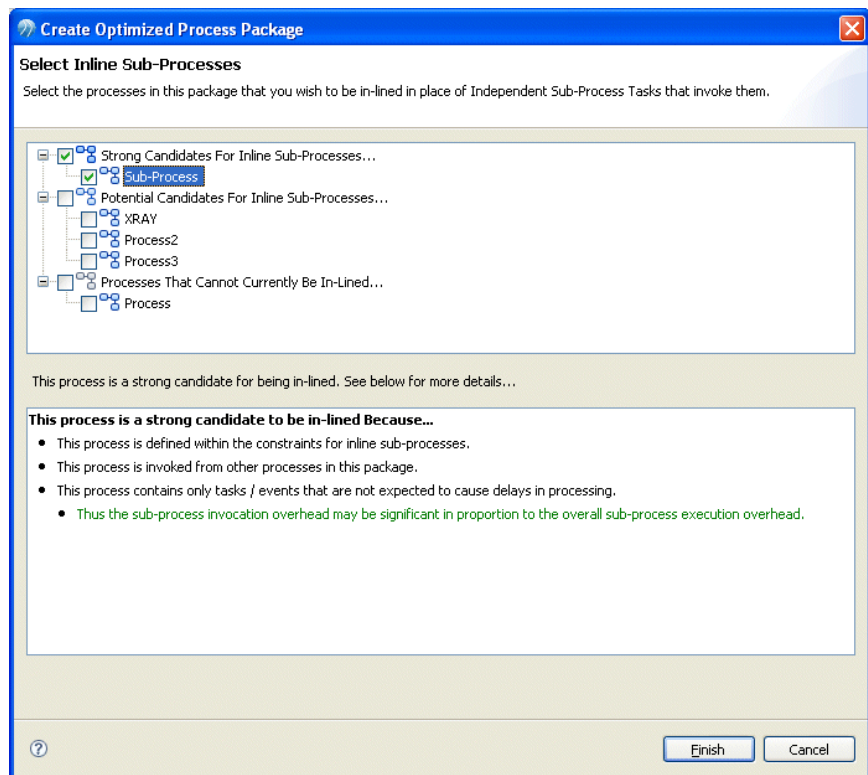
To create an inline sub-process, do the following:

1. In the Project Explorer, select the process that contains the reusable sub-process calls that you want to make inline.
2. In the Properties view, select the **Inline Sub-Process** option:



3. In the Project Explorer, right-click the package that contains the process, and select **Create Optimized Package**.

A dialog similar to the following is displayed:

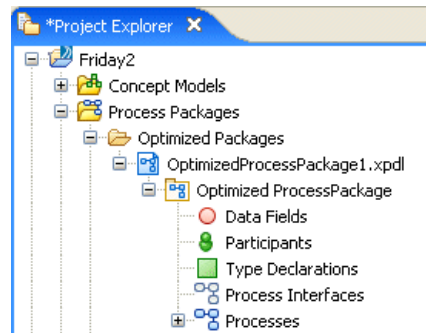


4. Select the sub-processes that you want to make inline and click **Finish**.



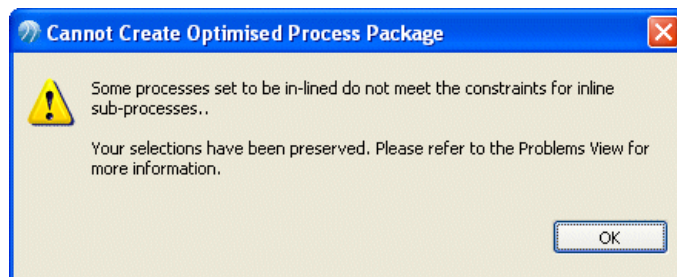
Clicking on the individual sub-processes provides information about whether the selected sub-process is a good candidate for conversion to an inline sub-process. For more information, see [Criteria for Inline Sub-Processes on page 215](#).

5. If there are no problems with the selected sub-processes, in the Project Explorer, a folder called **Optimized Packages** is created, and an optimized copy of the original process package is created:



In this example, the original process package was called **ProcessPackage1.xpdl**, and the optimized version becomes **OptimizedProcessPackage1.xpdl**.

If there are problems with the selected processes, the following message is displayed:



Click **OK**. The selections you made in the wizard are preserved, however before an optimized process package can be created you must correct the problems that were encountered.

Inline Sub-Process in Detail

This section describes in detail what happens to an inline sub-process (including its parameters, participants, and so on) when a package is optimized.

Process Objects

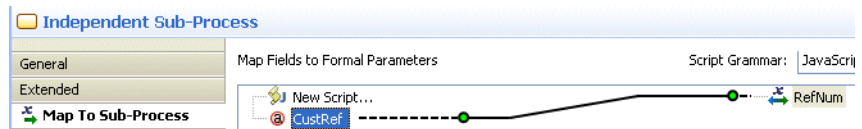
When a package is optimized, the following sub-process objects are moved into the main process, replacing the relevant sub-process calls:

- Associations
- Data Objects
- Events (except Start and End)
- Gateways
- Sequence Flow
- Tasks
- Text Annotations

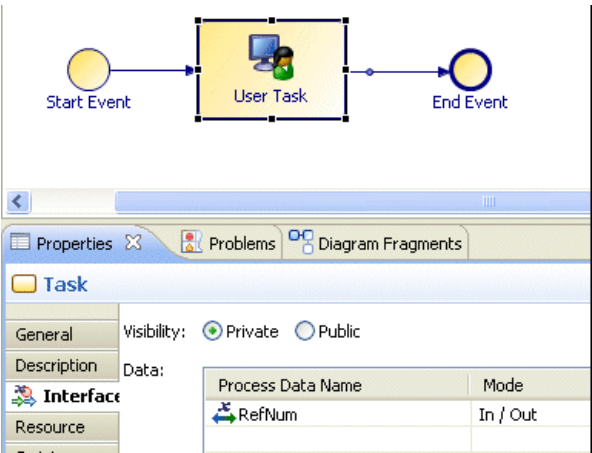
Start and end events are not moved into the main process.

Data Fields and Mapped Parameters

References to parameters in sub-processes are replaced according to the parameter mapping. For example, consider the parameter mapping to the following sub-process:



The data field **CustRef** in the parent process is mapped to **RefNum** in the sub-process. In the sub-process, the parameter **RefNum** is used by the single user task:



After refactoring as an inline sub-process, the user task is brought into the main process with **RefNum** as its parameter.



Every effort is made to ensure that when a sub-process is inlined, it is semantically the same as if it were not in-lined. However, not every potential scenario can be catered for.

For example, if the same calling-process field is used in the parameter mappings for two parallel reusable sub-process tasks, the calling-process field will now be used in-parallel in the two sets of sub-process contents that were moved up into the calling-process. This may have undesired effects.

Sub-Process Data Fields and Unmapped Formal Parameters

Sub-process data fields and unmapped formal parameters are moved up to the calling process after refactoring. In order to preserve the original semantics of the sub-process, any data fields or unmapped formal parameters in the sub-process that also exist in the calling-process are renamed (each new instance of the same named data is suffixed with a sequence number). Any references to the renamed sub-process data are updated when the sub-process activities and flows are moved up to the calling process.



Package-level data fields are not included in this behavior unless the sub-process is in a different package to the calling process.

Example

A sub-process has a data field called CustomerField which its activities use internally. The calling process also has a field called CustomerField which it uses internally.

When the sub-process is in-lined, the sub-process CustomerField is copied into the calling-process as CustomerField2. All references to CustomerField in sub-process activities and flows are replaced with references to CustomerField2.

Multiple Calls to a Sub-Process

This rule also applies when several sub-processes are called from a single process. If the sub-processes have the same data fields names, then each invocation will cause separate, sequence-numbered instances of data fields in the calling process when they are made inline.

Example

There are two calls to the same inline sub-process from a single calling process. The sub-process has a field called CustomerField.

The copy of sub-process activities and flows 'moved up' in place of one reusable sub-process task will operate on CustomerField and the other will operate on CustomerField2.

Field Name Conflicts

This rule also applies when inline sub-processes are nested, and have conflicting field names.

Example

The calling-process (MainProcess) calls an inline sub-process (SubProcess) which in turn calls a nested inline sub-process (SubSubProcess).

Each process has a field called CustomerField.

1. SubSubProcess is in-lined into SubProcess so that its instance of CustomerField becomes CustomerField2 in SubProcess.
2. SubProcess now has 2 fields, CustomerField and CustomerField2. When this is in-lined into MainProcess CustomerField in SubProcess is dealt with first (alphabetically) and is therefore renamed as CustomerField2 in MainProcess.
3. Now when the SubProcess field CustomerField2 is subsequently copied into MainProcess, a CustomerField2 already exists so it is renamed as CustomerField3.

Sub-Process Participants And Type Declarations

Participants are moved up to the calling process if the calling process does not already have a participant with the same name. If the Participant already exists in the calling process, references to the sub-process participant are exchanged for references to the calling process participant in the sub-process content that is moved up.

Similarly, type declarations (that are referenced by sub-process data fields and unmapped formal parameters) are moved up to the calling process if the calling process does not already have a type declaration with the same name. Otherwise, the calling-process type declaration is used.

Criteria for Inline Sub-Processes

When you right-click a package and select **Create Optimized Package**, the resulting wizard analyzes the package and reports on the package's suitability for optimization. It also reports problems in the Problems view.

Problems that Prevent Optimization

The following conditions result in errors in the Problems view and prevent optimization:

- The sub-process referenced by the task cannot be accessed.
- A reusable sub-process task generates an error if it has:
 - Boundary-attached events.
 - Multiple incoming sequence flows.
 - Multiple outgoing sequence flow.
 - A cyclic inline sub-process call.
 - Script parameter mappings.
- An inline sub-process generates an error if it has:
 - A call task set to multiple instance.
 - Multiple pools.
 - No end activity.
 - Multiple end points.
 - An end event not of type None.
 - An end event with multiple incoming sequence flow.
 - A nested invocation from external packages.
 - No start activity.
 - Multiple starting points.
 - A start event is not of type None.
 - A start event with multiple outgoing sequence flow.
 - No activities besides start and end events.
 - Inline sub-process in/out parameter '%s' must be mapped to the same calling process field.

Optimization Categories

As a result of the analysis by TIBCO Business Studio, packages fall into one of the following categories:

- [Strong Candidates For Inline Sub-Process](#)
- [Potential Candidates For Inline Sub-Process](#)
- [Processes That Cannot Currently Be In-lined](#)
- [External Package Inline Sub-Processes](#)

Strong Candidates For Inline Sub-Process

The following factors indicate that the process is a strong candidate for optimization.

- There are no problems that prevent optimization (see [Problems that Prevent Optimization on page 215](#)).
- The process is invoked from a reusable sub-process task in another process in the package.
- The process contains no tasks or events that cause a pause in the process flow (such as user/manual/receive task, intermediate event type message or timer). In these cases, process invocation overhead may be a significant portion of the overall sub-process execution overhead.

Potential Candidates For Inline Sub-Process

These packages do not have problems that prevent optimization, but contain processes with tasks or events that cause a pause in process flow. Such a package may still merit optimization. For example, the process may have user tasks, but if it is invoked frequently enough, there may be a performance improvement if the package is optimized.

Processes That Cannot Currently Be In-lined

These packages have problems that prevent them from being optimized (see [Problems that Prevent Optimization on page 215](#)), or are not invoked from a process within the same package (and would therefore disappear on optimization).

External Package Inline Sub-Processes

These are sub-processes in other packages that are already invoked from reusable sub-processes in the package being optimized.

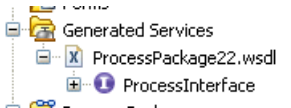
This category is for reference only (and to show if these sub-processes have problems).

These sub-processes are disabled in the wizard (you cannot de-select them).

Process Interfaces

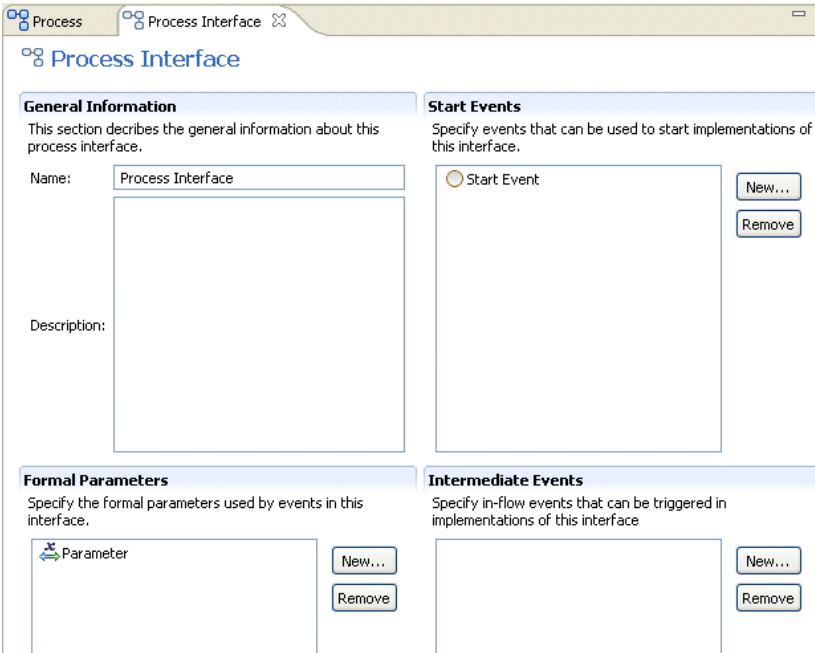
A process interface specifies the events and their parameters that must be present in processes created using that interface. It is the "contract" between the process and external applications that call the process. The use of a process interface allows the dynamic selection of sub-processes at runtime, promoting separation of the design-time and runtime environments.

Optionally, you can create a process interface with one or more message events. This creates an abstract WSDL file that is stored in a folder named Generated Services:



You can then call this abstract WSDL file in a service task, or optionally add the concrete implementation details to the WSDL file.

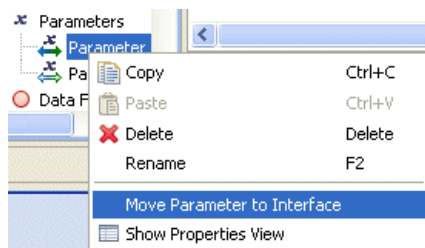
Using the Process Interface Editor, you can modify the interface to add events or parameters:



Once created, a process interface can be used by several different processes. If a process is created using a process interface, all the events and parameters specified in the interface must be present in a process that implements that interface. You can add additional events or parameters, but removing any of those required by the process interface will invalidate the process.



If you have additional parameters that are local to a process that implements a process interface, you can move the parameters into the process interface by right-clicking the parameter and selecting **Move Parameter to Interface**:



This option is only available for processes that implement a process interface in the same package, and only for parameters that do not have problems in the Problems view.

If you create a process using a process interface, the process that you create inherits the events and parameters created in the interface. For example, a message event in a process created using a process interface displays the following properties:



If you cannot see the right-hand side of the following dialog, switch to the Solution Design capability using the following menu or click **Provide Implementation Details**.



The screenshot shows the 'Intermediate Event' properties dialog. The 'General' tab is selected. The 'Implements Method' is 'Intermediate Event'. The 'Trigger Type' is 'Message'. The 'Implementation' is 'Web Service'. The 'Operation' is 'IntermediateEvent'. The 'Port Type' is 'bankxmlInterface'. The 'Port Name' is empty. The 'Service Name' is empty. The 'Transport' is 'SOAP over HTTP'. The 'Message Name' is empty. The 'To' and 'From' fields are empty. The 'Fault name' is empty.

Properties	Problems	Diagram Fragments
Intermediate Event		
General	Implements Method: Intermediate Event	Implementation: Web Service
Description	Trigger Type: Message	Operation: Select Clear Import WSDL
Interface	Message Name:	Port Type: bankxmlInterface
Resource	To:	Operation Name: IntermediateEvent
Scripts	From:	Port Name:
Appearance	Fault name:	Service Name:
Extended		Transport: SOAP over HTTP
Advanced		

Creating a Process Interface

To create a process interface, do the following.

1. Select **File > New > Other**.
2. Expand **Business Modeling > Business Process Modeling**, select **Process Interface**, and click **Next**.
3. Enter a name for the process interface you are creating and click **Next**.
4. Enter a description for the process interface you are creating and click **Next**.
5. Select the intended destination environments for the process interface and click **Finish**. The Process Interface editor is displayed. Use this to add or remove start events, intermediate events, and parameters.
6. Enter any extended attributes and click **Next** (for more information, see [Creating Extended Attributes on page 51](#)).

Parts of a Process Interface

A process interface consists of start events, intermediate events, and their associated formal parameters and errors.

Start Events

A process interface must have one or more start events (one is automatically created by default) that specify how a process instance is initiated.

When more than one start event is specified, a process implementing the interface can be invoked by any of the start events specified in the interface. As per BPMN, multiple separate start events are considered exclusive (each starts a new process instance).

The start event for a process interface can be of type None or Message:

- A maximum of one start event of type None can be specified. Use this type of start event for invoking a sub-process using a reusable sub-process task or when a sub-process is meant to be manually invoked. See [Using a Dynamic Sub-Process on page 232](#).
- Use message start events for the invocation of the process using a message based interface (for example, a web service operation).

- A start event for a process interface can be associated with mandatory formal parameters using the **Interface** tab. For example:

General

Description

Interface

Advanced

Visibility: ☐ Private ☒ Public

Parameters:

Parameter Name	Mode	Description	Mandat...
FP1	In / Out		☒

If a process instance is invoked using this start event, the parameter **FP1** must be specified.

Both input and output (including combined in/out) parameters can be associated with the event. Specifying output parameters means that these are the parameters should be returned when the process is invoked using this start event.

Intermediate Events

The intermediate events for a process interface can be of type None or Message:

- **None** Use this type of intermediate event for manual event triggering.
- **Message** Use this type of intermediate event for triggering using a message based interface (for example, a web service operation).

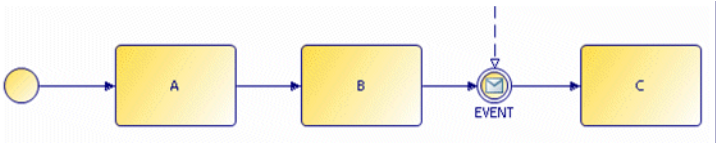
An intermediate event can be associated with input or input/output formal parameters that should be present when triggering the event, as well as with errors that will be translated into catch error end events in implementing processes.



Output formal parameters cannot be associated with catch message intermediate events.

Event Persistence

Because a process interface does not specify the intended location of the event within the flow of the process implementation, events are considered "persistent". As such, they can be triggered at any point in the process flow that is upstream of the event, or when the flow is paused waiting for the event to be triggered. For example:



The trigger for the intermediate event can be received while Task A or B is being processed, or when the flow is paused at the intermediate event waiting for the event to be triggered. In either case, Task C is not processed without an event trigger. If the trigger is received during processing of Task A or B, the trigger arrival is persisted, and the event is triggered immediately when it is reached in the flow.

Errors

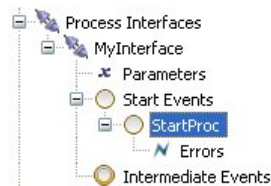
TIBCO Business Studio allows you to specify errors that may be thrown by a process that implements the process interface. This is useful where:

- The process interface defines a messaging or web-service interface. In this case, the error events are translated into WSDL operation fault messages.
- The process interface is used for dynamic sub-process invocation. In this case, the error events are translated into end error events in process implementations. Independent sub-process tasks that reference process interfaces can then identify the thrown error events directly from the process interface.

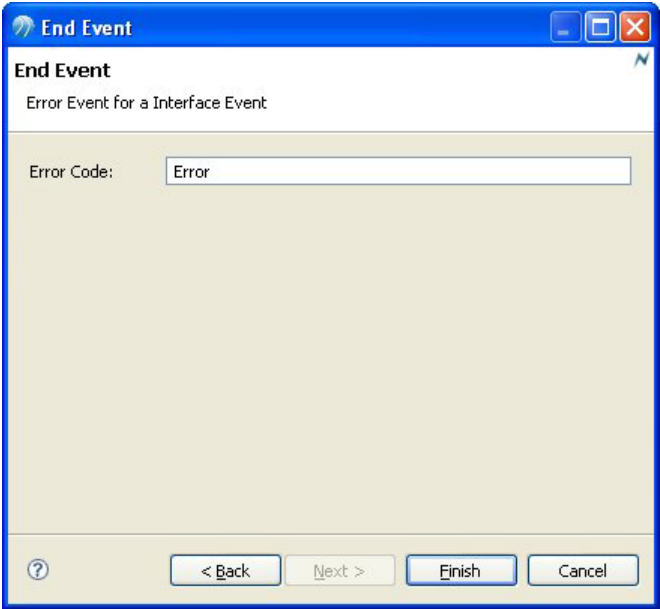
Creating Error Events

Error events are created in the Project Explorer as follows:

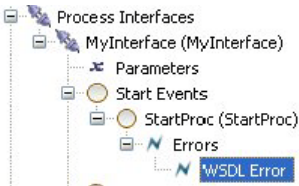
1. Create a process interface.
2. In the Project Explorer, under the process interface, expand an event. For example:



3. Right-click **Errors** and select **New > Error**. The following dialog is displayed:



4. Enter a name for the error code and click **Finish**. The error code you created is displayed in the Project Explorer, and in the Process Interface Editor:



Continue with [Error Codes and Process Interfaces on page 225](#) to see how these error codes are used in processes created from a process interface with error codes.

Error Codes and Process Interfaces

When a process implements a process interface that has error codes, the error codes can be used to throw either WSDL fault messages from a message-based process, or errors thrown by a sub-process.

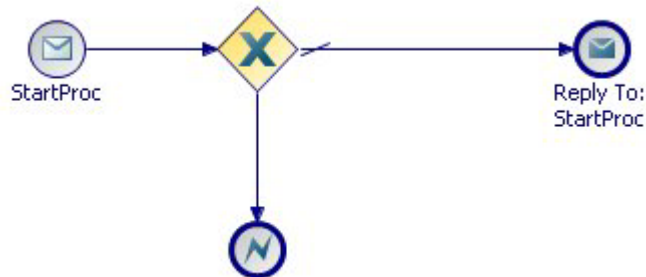
For example, consider the following process interface:

The screenshot shows the 'BALInt' process interface configuration window. It is divided into four main sections:

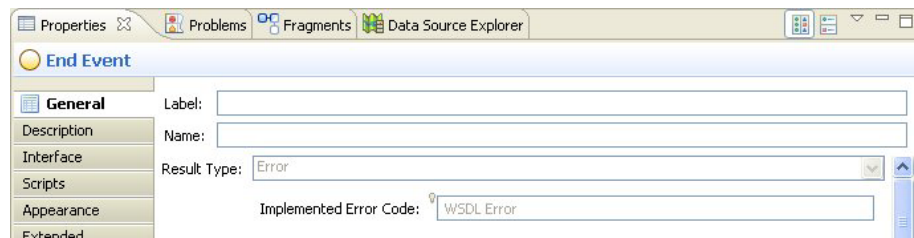
- General Information:** Contains fields for 'Name' (BALInt) and 'Description' (empty).
- Start Events:** Contains a list of events: 'StartProc (StartProc)', 'Interface Parameters', 'Errors', and 'WSDL Error'. There are 'New...' and 'Remove' buttons.
- Formal Parameters:** Contains a list of parameters: 'Account (Account)' and 'Balance (Balance)'. There are 'New...' and 'Remove' buttons.
- Intermediate Events:** Contains a list of events (empty). There are 'New...' and 'Remove' buttons.

This process interface has a start event, an error code (**WSDL error**), and two parameters, **Account** and **Balance**.

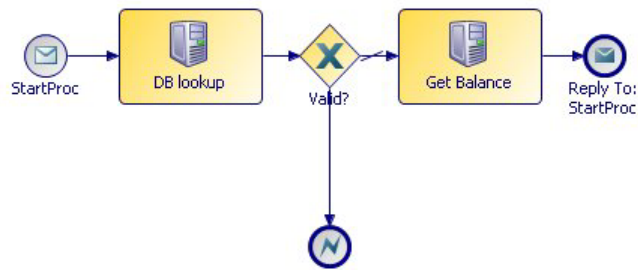
When you create a process from this interface, it creates the following:



The default sequence flow from the gateway replies to the start event. The conditional sequence flow from the gateway throws an error with the code **WSDL error**.



The process with more tasks looks like this:

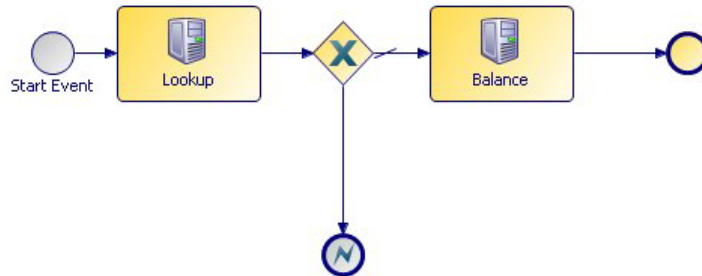


The process flow is as follows:

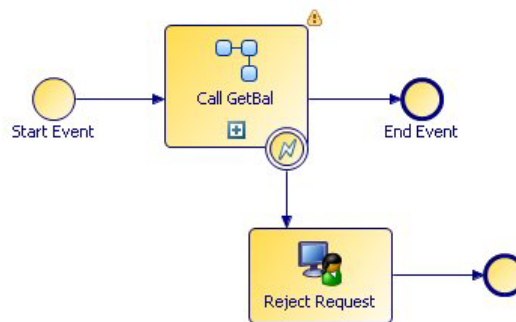
1. The process is started upon receipt of a message that contains an account.
2. The database task looks up the account in the database to verify if it is valid.
3. If the account is valid, another database task gets the balance, and the balance and account number are sent as output.

If the account is not valid the error code is thrown.

The following example shows how the previous "get balance" example could be implemented as a sub-process (in this case, the process interface has normal start and end events rather than start message and end message events):



In this case, the sub-process throws an error, and in the calling process the error is dealt with:



The catch error event is configured to catch the error thrown from the sub-process:

Intermediate Event	
General	Label:
Description	Name:
Interface	Trigger Type: Catch Error
Scripts	Catch Error Code: InvalidAcct
Map From Error	Thrown By: Call GetBal / BalSub / <Error End Event>
Appearance	
Extended	

Making a Sub-Process Call to a Process Interface

There are two different ways of creating a call to a process interface:

- By dragging a process interface from the Project Explorer and dropping it onto your process.
- Using the reusable sub-process tool from the palette.

Drag and Drop

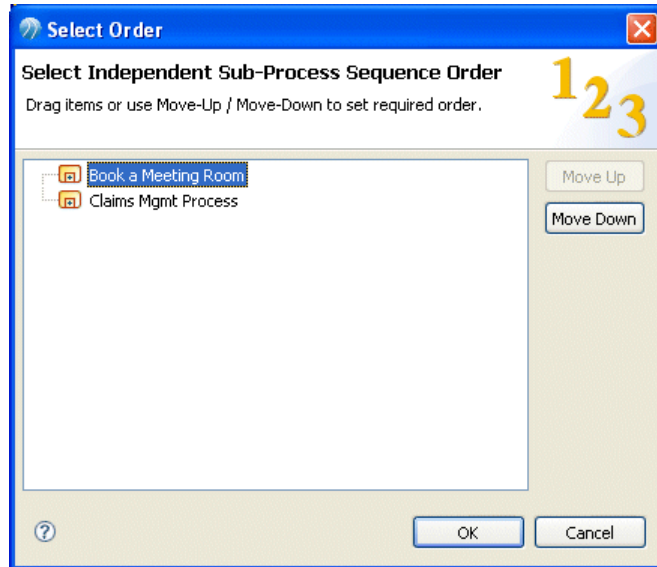
To create a reusable sub-process call using drag and drop, do the following:

1. Expand the Project Explorer to locate the process interface that you want to implement.
2. Click the process interface, holding down the mouse button, drag the pointer to the calling process (open in the Process Editor), and release the mouse button.



You can select multiple processes for drag and drop operations using the Ctrl key.

3. If you are dropping more than one process interface, a menu is displayed with two options:
 - **Create Sub-Process Task Sequence** Selecting this option allows you to create sub-process tasks connected by sequence flow. The following dialog is displayed to allow you to control the order of the tasks:



Use the **Move Up** and **Move Down** buttons to control the order of the tasks. When you are finished, click **OK** to place the tasks.

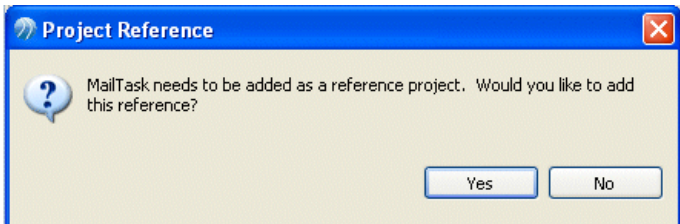
- **Create Unsequenced Sub-Process Tasks** Selecting this option places the tasks in the process without a connecting sequence flow.

Palette To create a reusable sub-process call process interface using the palette, do the following:

1. In the Process Editor, select the **reusable sub-process** tool.
2. Click in the process where you want to place the activity that calls the sub-process.
3. On the Properties view for the activity, browse for the process interface you want to call as a sub-process.

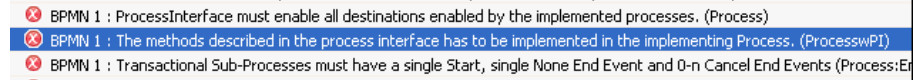


If the process interface you select is not in the current project, you are prompted to create a project reference:



Modifying a Process Interface

You can modify a process interface using the Process Interface editor, however any changes made cause validation errors in processes that have already implemented the interface. For example:

- 
- ✖ BPMN 1 : ProcessInterface must enable all destinations enabled by the implemented processes. (Process)
 - ✖ BPMN 1 : The methods described in the process interface has to be implemented in the implementing Process. (ProcesswPI)
 - ✖ BPMN 1 : Transactional Sub-Processes must have a single Start, single None End Event and 0-n Cancel End Events (Process:Er

To correct this problem, ensure that the changes made to the process interface are reflected in any processes that have already implemented the interface.

Using a Dynamic Sub-Process

Dynamic sub-processes are used when a process calls one of several sub-processes at runtime, but it is not known at design time which sub-process will be called. The exact sub-process to be run is chosen at runtime, depending on the process data.

In order to dynamically choose the sub-process to execute at runtime it must obey an interface that is known at design-time. This is provided by a process interface which is selected in a Reusable Sub-Process task instead of an actual sub-process type.

Creating a Dynamic Sub-Process

To create a dynamic sub-process:

1. Create a process interface that specifies the start event and its input/output parameters. Each process that is to be invoked from the dynamic sub-process task **must** implement the same process interface. See [Creating a Process Interface on page 221](#).
2. Create one or more Reusable Sub-Process tasks in your process.
3. From the General tab for each Reusable Sub-Process task:
 - a. Using the picker, () select the process interface created in [step 1](#).
 - b. In the Runtime Identifier Field select a formal parameter or data field using the picker (). This must be a text field or an array. Arrays can be used for multi-instance sub-process tasks where potentially different sub-processes are required for each instance of the same task.

☐ Is A Transaction

Invocation of Sub-Process: ([Open Sub-Process](#))

Sub-Process location:

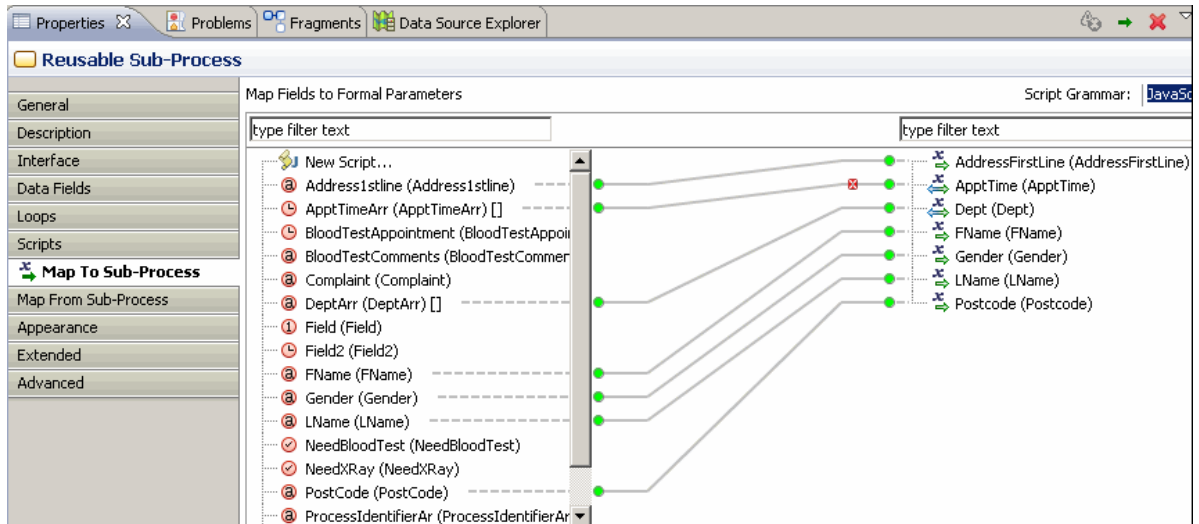
Sub-Process name:

Runtime Identifier Field:

4. Initialise the Runtime Identifier Field. For example, for a multi-instance sub-process task you could create a Script task before it to populate the array data field with a sub-process name element for each task instance. For a single instance sub-process task simply set the runtime identifier field to the required sub-process name in a script prior to the task.

5. In the Reusable Sub-Process task, map data to and from the called process (to pass data between the process and sub-process):
 - Map the input to the interface in Map To Sub-Process.
 - Map the output from the interface in Map From Sub-Process.

For example:



Automatically Passing Different Data To and From Sub-Process Instances

If the Reusable Sub-Process task is multi-instance (or loop) it is possible to automatically pass different data to and from each separate sub-process instance.

Inputting Different Data To Each Sub-Process Instance

To input different data to each sub-process instance:

1. Specify an array data field that matches the type of a non-array sub-process/interface input parameter.
2. In a script prior to the sub-process task, populate this array data field with different data for each sub-process instance. For example:
 - `appointmentTypeArray.set(0, "X-Ray");`
 - `appointmentTypeArray.set(1, "PlasterDept");`
 - `appointmentTypeArray.set(2, "Physio");`
3. In the "Map to Sub-Process" property tab, map the array data field to the non-array sub-process/interface parameter. The sub-process invoked from

each instance of the task will receive the list element corresponding to the instance index of that task (see `getActivityLoopIndex()` - refer to the the 'Script Functions' appendix in the *TIBCO Business Studio BPM Implementation Guide* for more information). For example:

- The first instance (activity loop index=0) will receive the data "X-Ray" into its "appointmentType" parameter.
- The second instance (activity loop index=1) will receive the data "PlasterDept" into its "appointmentType" parameter.
- The third instance (activity loop index=2) will receive the data "Physio" into its "appointmentType" parameter.

Returning Different Data From Each Sub-Process Instance

To return different data from each sub-process instance:

1. Specify an array data field that matches the type of a non-array sub-process/interface output parameter.
2. In the "Map From Sub-Process" property tab, map the output parameter to that array field.

At runtime the array field will be populated with the return data from each sub-process instance. The list element index will correspond to the activity loop index from which the sub-process instance was invoked.

For example, if the sub-process returned an integer parameter "ReturnParam" with value "X-OK" when passed "X-Ray", "PL-OK" when passed "PlasterDept" and "PH-OK" when passed "Physio" and you mapped this to an array data field "ReturnArrayField" then:

- `ReturnArrayField.get(0)` will be "X-OK"
- `ReturnArrayField.get(1)` will be "PL-OK"
- `ReturnArrayField.get(2)` will be "PH-OK"

The way that this data is passed behaves in the same way for multi-instance statically defined sub-process tasks (tasks that reference an actual sub-process at design time).

Chapter 10 **Integrating with Other Applications**

This section describes use TIBCO Business Studio to integrate with other applications.

Topics

- [Destinations and Validation, page 236](#)
- [Integrating with ARIS Software, page 240](#)
- [Object Mappings — ARIS Software, page 244](#)
- [Integrating with Other Software, page 247](#)

Destinations and Validation

When you specify a *destination environment* for a Process, you are specifying the intended runtime environment in which the Process will execute. According to the destination environment that you specify, TIBCO Business Studio performs validation on the Process.

Any error messages resulting from this validation will be displayed in the Problems view and indicate which parts of your Process need to be changed.

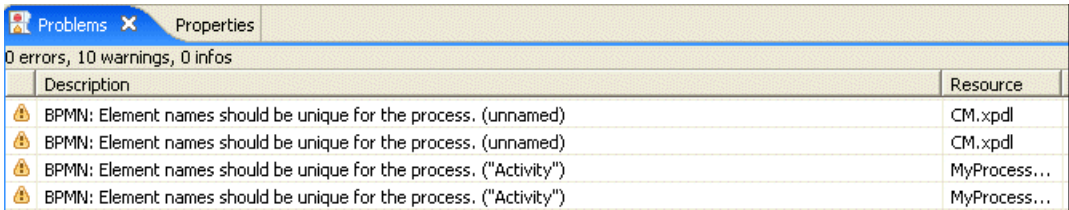
You can create your own destination environments, and the specific "destination components" that make up these destination environments can be customized in the Preferences. To view the current configuration of destination environments, select **Window > Preferences**, and select **Destination Environments**. For more information about customizing destination environments, see the destination-specific implementation guide.

Correcting Validation Errors

Any problems that result from validation are shown in the Problems view. To correct the problem do one of the following:

- Right-click the problem and select **Quick Fix** (if enabled for the current problem). The gives you the option of having TIBCO Business Studio correct the problem for you.
- Double-click the problem or right-click the problem and select **Go To**. This displays the Process in the Process Editor, highlighting the offending object and allowing you to manually correct the problem.

Example



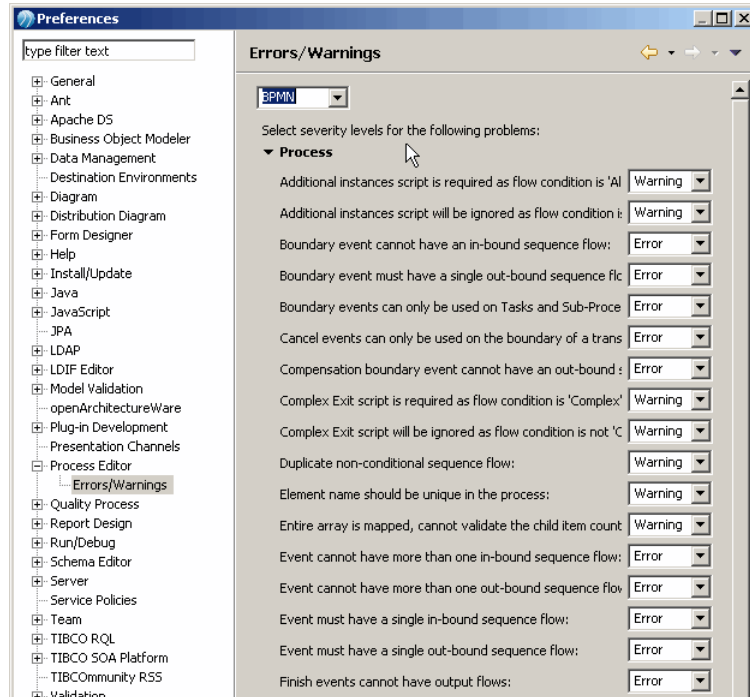
The screenshot shows the 'Problems' view in TIBCO Business Studio. The title bar includes 'Problems' and 'Properties'. Below the title bar, it states '0 errors, 10 warnings, 0 infos'. The main area is a table with two columns: 'Description' and 'Resource'. There are four rows of warnings, each preceded by a yellow warning icon. The first two rows have the same description: 'BPMN: Element names should be unique for the process. (unnamed)' and both point to 'CM.xpdl'. The third row has the description 'BPMN: Element names should be unique for the process. ("Activity")' and points to 'MyProcess...'. The fourth row has the same description and points to 'MyProcess...'.

Problems	
Properties	
0 errors, 10 warnings, 0 infos	
Description	Resource
BPMN: Element names should be unique for the process. (unnamed)	CM.xpdl
BPMN: Element names should be unique for the process. (unnamed)	CM.xpdl
BPMN: Element names should be unique for the process. ("Activity")	MyProcess...
BPMN: Element names should be unique for the process. ("Activity")	MyProcess...

Setting the Validation Preferences

You can customize the validation that is performed in the Process Editor. Specifically, for each validation error you can specify its severity level as **Error**, **Warning**, **Info**, or **Ignore**. To customize the Process Editor validation, do the following:

1. Select **Window > Preferences**.
2. Expand **Process Editor** and select **Errors/Warnings**. The following dialog is displayed:



3. If you want to change the severity level for a Process Editor problem, select from the drop-down list. When you have finished, click **Apply** to effect any changes you have made.



You can downgrade the severity of BPMN errors, however some TIBCO Business Studio errors are not displayed because downgrading them would allow processes to be deployed that would be invalid in the runtime environment.

4. The following dialog is displayed:



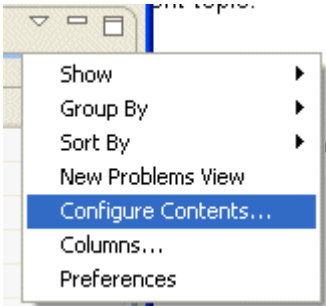
- Click **Yes** to revalidate your workspace. Depending on the size of the workspace and the number of errors, there is a delay while the revalidation occurs.
- Click **No** to revalidate your workspace later. The revalidation will take place when the concept file next changes or is saved, or when you explicitly request a rebuild of the project or workspace.
- Click **Cancel** if you do not wish to apply your changes.



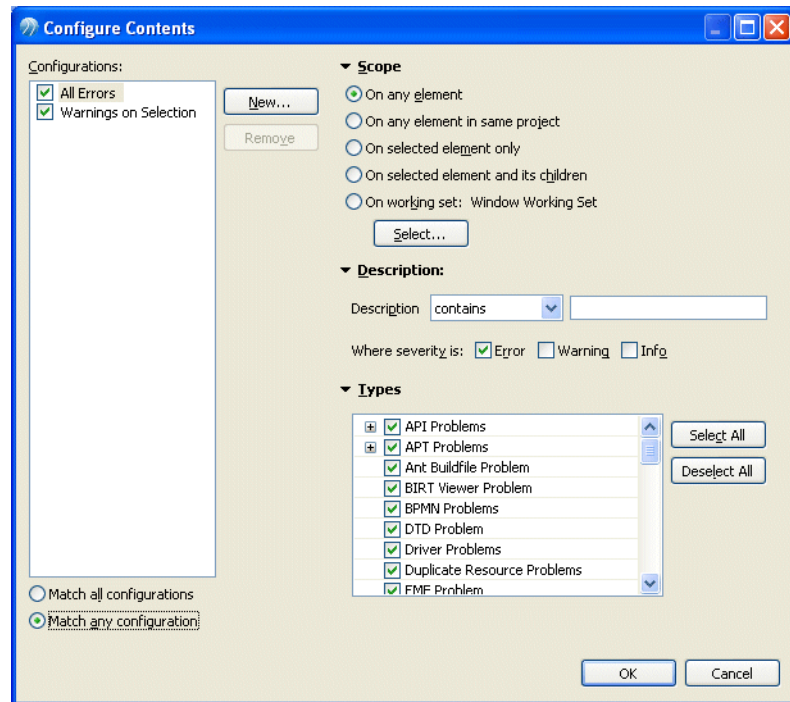
At the top of the dialog there is a drop-down list that allows you to select either **BPMN** or **Simulation**.

Configuring the Problems View

You can configure the appearance and display of the Problems view by selecting items from the menu in the upper right of the Problems view:



By selecting **Configure Contents**, you can filter the problems that are displayed:



Two configurations are provided by default, **All Errors** and **Warnings on Selection**. You can create and save your own configurations. You also can reduce the number of errors displayed by selecting a different scope. For example, by choosing **Selected element only** as the scope in conjunction with the **All Errors** configuration, you will see all errors for the selected element only rather than for all elements in the workspace.

By default, problems in the Problems view are grouped by severity of the problems. However, you can change this using the Group By menu option. For example, to arrange the Problems by their type, select **Group By > Plug-in Problem Type**. You can also add or remove types in the previous **Configure Contents** dialog.

Integrating with ARIS Software

ARIS software provides business diagramming and analysis capabilities. You can import processes created with ARIS 6.2, 7.0 and 7.1 software into TIBCO Business Studio. First, use the XML Export Wizard in ARIS to create an XML file. Then, use the Import Wizard in TIBCO Business Studio to import the XML into TIBCO Business Studio.



- Only eEPC (Extended event-driven process chain) type processes and associated FADs (Function Allocation Diagrams) created using the ARIS Toolset can be imported.
- If you have selected processes from several different groups in ARIS, they are imported under one Package in TIBCO Business Studio.

For more information, see [Object Mappings — ARIS Software](#) on page 244.

Creating Custom Symbols in ARIS (Optional)

You can create custom symbols in ARIS processes that you want to export. For example, you can specify that a certain type of ARIS object displays a graphic of your choice rather than the default graphic.

1. Log in to the ARIS database that contains the data you want to export.
2. Expand **Configuration** and click **Method**.
3. Click the **Symbols** tab.
4. Right-click the object for which you want to specify a different symbol and select **New > Symbol**.
5. Browse to select a graphic of the type AMF, WMF, or EMF.
6. Click **Next** and change the **Name** of the object if desired.
7. After you have created the object, log out of the database.
8. Choose **Configuration > Update Configuration**.
9. Log in to the ARIS database.

Using Custom Symbols in TIBCO Business Studio

If you have created custom symbols in ARIS, you must perform the configuration described in this section before you can use those symbols with ARIS processes that you import.

1. Navigate to the `\LocalServer\sysconfig` subdirectory of the ARIS installation directory.
2. Open the **methodextension.cpf** file. This file contains a universal ID (**Guid**) for each custom symbol that you have defined. For example:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<!-- Do not edit this file !!-->
<SCSMethodExtension version="1" >
  <UserDefName KindNum="1" ItemTypeNum="66218" LocaleId="1033"
    Name="Actor (sample)"/>
  <MetaFile Guid="da459755-894d-4a86-86bb-66624fde7434"
    NewSN="65549" RefSN="13" size="12036" Amf="false"/>
  <MetaFile Guid="b783fa5d-a484-46f6-a357-b848ef6e328c"
    NewSN="65569" RefSN="33" size="12036" Amf="false"/>
  <MetaFile Guid="9616e674-0dcb-40ac-b370-e7bc28271949"
    NewSN="66218" RefSN="682" size="12036" Amf="false"/>
</SCSMethodExtension>
```

The **Guids** listed in the file correspond to graphics files located in the same directory.

3. Locate the graphics files listed in the **methodextension.cpf** file and copy them to a temporary directory.
4. Rename the graphics files to **.emf** or **.wmf** file extensions.
5. Open each image and save it to Portable Network Graphics (PNG) format.
6. In TIBCO Business Studio, create a folder called **images** in the Project into which you will import the ARIS Process.
7. Copy the PNG versions of the graphics into the **images** folder.
8. Export the ARIS process and then import it into TIBCO Business Studio as described in this chapter. The custom symbols you created in ARIS should be displayed in TIBCO Business Studio.



If the symbol does not display correctly, click the Data Object and check that the file specified for **External References** is correct.

Exporting from ARIS

Export ARIS data to XML as described in this section.

Before Exporting from ARIS

If any of the Processes or groups of Processes that you want to export contain FADs, do the following:

1. Log in to the ARIS database that contains the data you want to export.

2. Open the main process.
3. Select the function type that contains the FAD.
4. Right-click and choose **Select > Select All Objects of the 'Function' Type**.
5. Right-click the selection and select **Assignments > Show**.
6. Save the process and perform the export as described in the following section.

Exporting from ARIS

To export the ARIS data, do the following:

1. Log in to the ARIS database that contains the data you want to export.
2. Highlight the group or Business Processes that you want to export.
3. Right-click and select **Export/Import > XML Export**. The **Select Attribute Language** dialog is displayed.
4. Select the language and click **Next**. The **Select Export File** dialog is displayed.
5. Browse for the location for the Export.
6. Click **Finish** to complete the export.

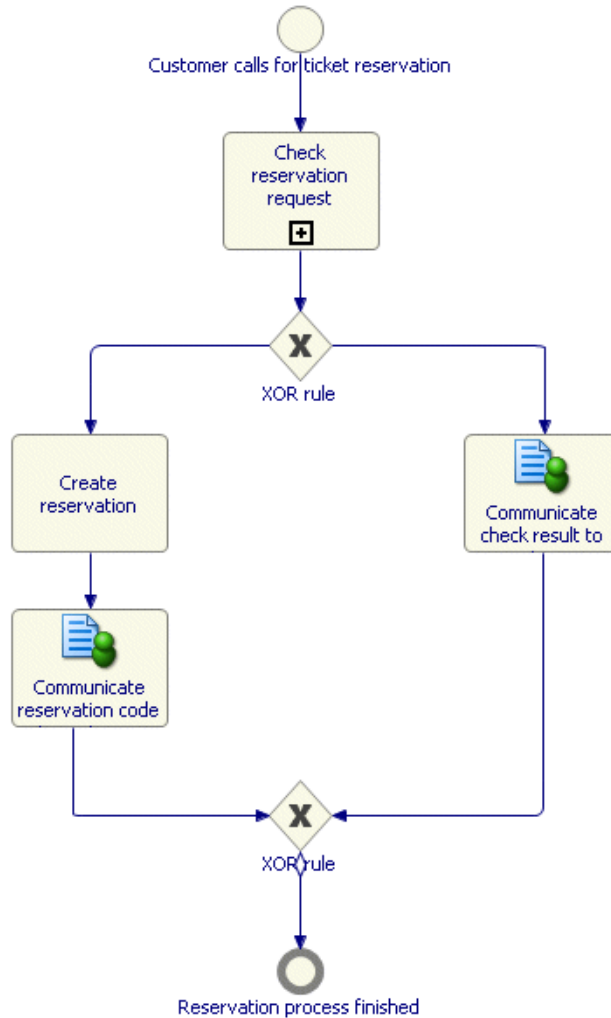
Importing ARIS XML Files into TIBCO Business Studio

To import an ARIS XML file into TIBCO Business Studio, do the following:

1. Select **File > Import**. The Import Wizard is displayed.
2. Expand **Business Process Management**, select **ARIS XML** and click **Next**.
3. Browse for the Directory where the ARIS XML file is located.
4. Browse for the Project folder into which you want to import the ARIS XML file and click **Finish** to complete the import.

Example

In ARIS Version 6.2, in the **Demo62** database, there are several examples of eEPC processes. The following example shows the TIBCO Business Studio import of the ARIS example process **Ticket Reservation** (in **Movie Palace > Business processes > Customer service > Ticket reservation**).



Object Mappings — ARIS Software

When you import a Process from ARIS Software into TIBCO Business Studio, the original objects from ARIS are mapped into objects that TIBCO Business Studio supports. This section describes the mapping between ARIS objects and TIBCO Business Studio objects.

Table 3: ARIS and TIBCO Business Studio Object Mappings

ARIS Object	ARIS XML Type	Corresponding TIBCO Business Studio Object
Event 	• (Start) OT_EVT with outgoing connections only • (End) OT_EVT with incoming connections only • (Intermediate) OT_EVT with incoming and outgoing connections	Depending on the position of the Event, a Start Event, Intermediate Event, or End Event.
Function (User Activity type; for example )	OT_FUNC, not starting with ST_SYS	User Task.
Function (System Activity type; for example )	OT_FUNC, starting with ST_SYS	Service Task.
AND Rule 	OT_RULE with: ST_OPR_AND_1, ST_OPR_AND_2, ST_OPR_AND_3, ST_AND	Parallel Gateway.
XOR Rule 	OT_RULE with: ST_OPR_XOR_1, ST_OPR_XOR_2, ST_OPR_XOR_3, ST_XOR	Exclusive Data Based Gateway.
OR Rule 	OT_RULE with: ST_OPR_OR_1, ST_OPR_OR_2, ST_OPR_OR_3, ST_OR	Inclusive Gateway.

Table 3: ARIS and TIBCO Business Studio Object Mappings (Cont'd)

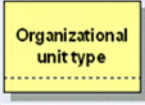
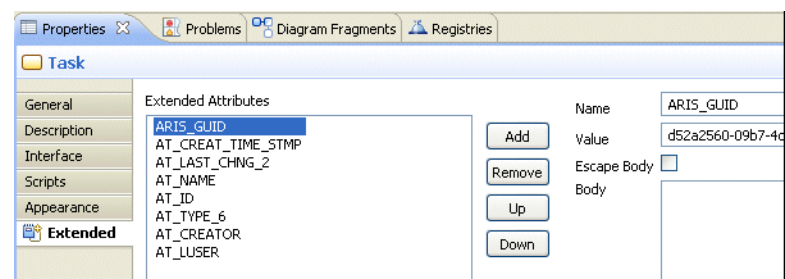
ARIS Object	ARIS XML Type	Corresponding TIBCO Business Studio Object
Rule 	OT_RULE (with none of the previous SymbolNum types)	Default (Exclusive Data Based) Gateway.
Organizational Unit Type, Organizational Unit, System Organizational Unit, and so on.  	OT_ORG_UNIT, OT_ORG_UNIT_TYPE	Participant of the type Organizational Unit (multiple Participants are supported).
Person Type, Internal Person, External Person, Group	OT_PERS, OT_PERS_TYPE, OT_GRP, OT_EMPL_INST	Participant of the type Human (multiple Participants are supported).
Position Type, Position, Position Description	OT_POS	Participant of the type Role (multiple Participants are supported).
Application System Type, System Organizational Unit, Module Type	OT_SYS_ORG_UNIT, OT_SYS_ORG_UNIT_TYPE, OT_APPL_SYS, OT_APPL_SYS_CLS, OT_APPL_SYS_TYPE, OT_MOD_TYPE, OT_MOD	Participant of the type System (multiple Participants are supported).
D Attribute (ERM), K Attribute (ERM), FK Attribute (ERM), COT Attribute, Technical Term 	OT_ERM_ATTR, OT_TECH_TRM, OT_ATTR_TYPE_GRP	Data Field.
Non-function object without connections		Data Object.

Table 3: ARIS and TIBCO Business Studio Object Mappings (Cont'd)

ARIS Object	ARIS XML Type	Corresponding TIBCO Business Studio Object
Non-function object that is not a recognized data type or participant and has only one connection (incoming or outgoing) to a function object		Data Object with connection.

Extended Attributes

Any attributes of an ARIS XML file that are not required or recognized by the TIBCO Business Studio Process model are stored as extended attributes. This preserves the information after the import into TIBCO Business Studio. For example:



Integrating with Other Software

You can integrate with the following software. The links provided are to additional information and files you will need which are provided on the Community forum.

- BizAgi Business Process Management software
<http://www.tibcommunity.com/docs/DOC-1964>
- Savvion Business Process Management software
<http://www.tibcommunity.com/docs/DOC-1962>
- IBM Rational System Architect Import Wizard
<http://www.tibcommunity.com/docs/DOC-1981>
- Mega 2009
<http://www.tibcommunity.com/docs/DOC-2001>

Chapter 11 **Reference**

This section describes the major parts of the TIBCO Business Studio user interface, including the perspectives and views that are provided.

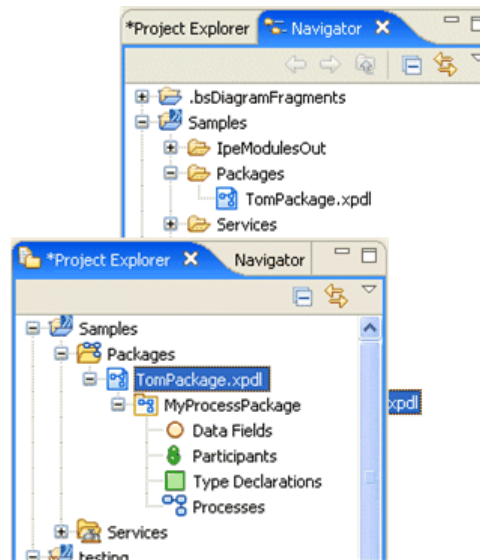
Topics

- [Project Explorer View, page 250](#)
- [Problems View, page 251](#)
- [Outline View, page 252](#)
- [Fragments View, page 253](#)
- [Properties View, page 254](#)
- [Cheat Sheets View, page 266](#)
- [Process Editor, page 267](#)

Project Explorer View

This shows a view of the project in which you can view and create projects, packages, processes and assets such as services and so on. In contrast to the Navigator view (available within the Resource Perspective) the Project Explorer view shows a contextual view of the Project.

For example, the following shows both the Navigator and Project Explorer views of the same Project.



To see this view, select **Window > Show View > Project Explorer**, or **Window > Show View > Other**, then expand the **General** folder and select **Project Explorer**.

You can also use the search facility to locate workspace elements in the Project Explorer view such as processes, package names, and process interfaces by pressing **Ctrl+F** or the search toolbar button in the Project Explorer view. For more information, see [Finding Objects on page 25](#).

Problems View

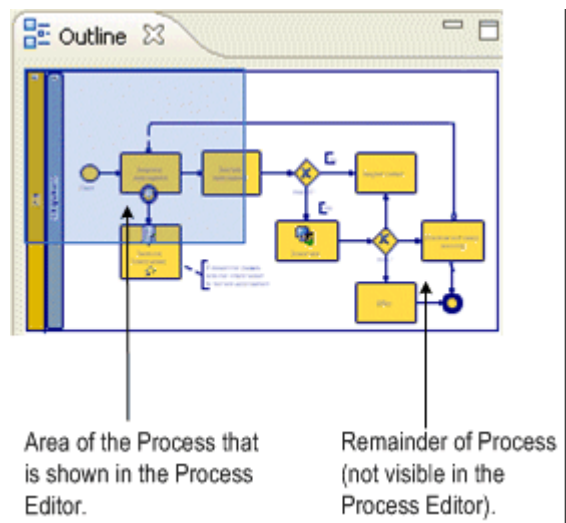
The Problems view is where the software displays warnings, errors and messages about the resources used in your Project. To see this view, select **Window > Show View > Problems**, or **Window > Show View > Other**, then expand the **General** folder and select **Problems**.

The Problems view also displays error messages that result from validation of your process. When you save your process, it is validated for BPMN correctness and for deployment or export to the selected destination environments.

For information about correcting problems, see [Correcting Validation Errors on page 236](#).

Outline View

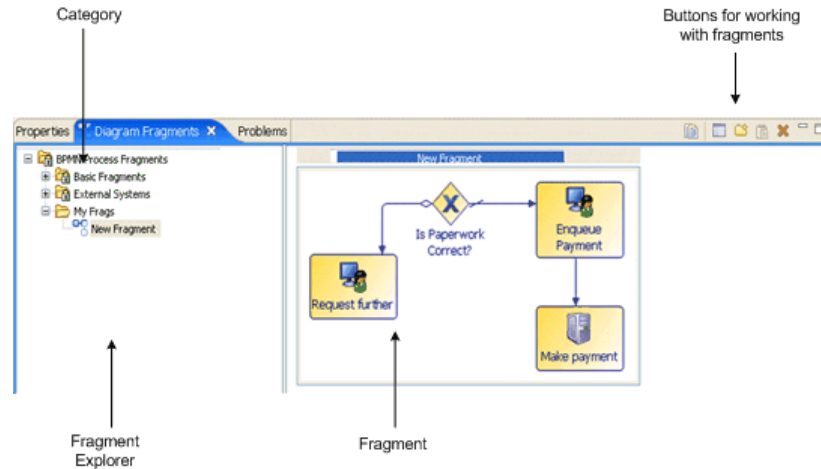
The **Outline View** allows you to focus on specific areas of a large process. The **Outline View** looks like this:



To see this view, select **Window > Show View > Outline**. By positioning the mouse pointer over the shaded area and holding the left mouse button down, you can reposition the shaded area of the **Outline View** and thus see different parts of the process.

Fragments View

The **Fragments** view allows you to store selected objects from your process and reuse them in a different process. You can also select predefined process fragments and use them in your process:



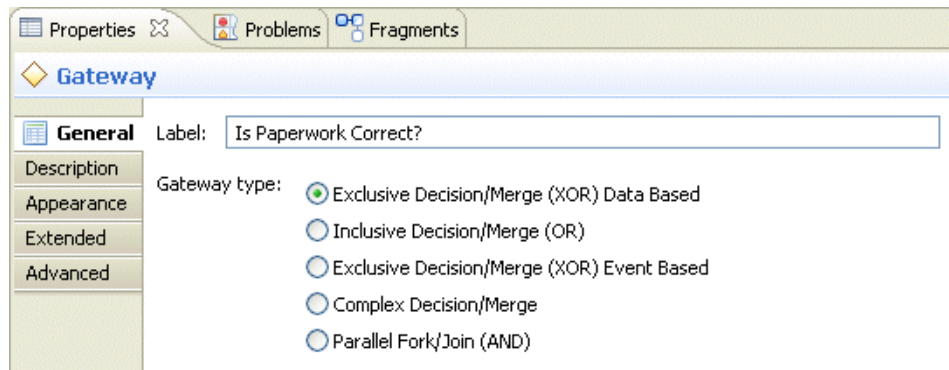
To see this view, select **Window > Show View > Fragments**. The buttons in the upper right of the view allow you to do the following:

-  - Copy a diagram fragment to the clipboard in preparation for pasting it onto a Process.
-  - Edit fragment/category properties such as the **Name** and **Description**.
-  - Create a custom fragment category in which to store your fragments.
-  - Paste a fragment from clipboard into the Fragments view or into the process.
-  - Delete a fragment or category.

Properties View

The **Properties View** shows you detailed information about the currently selected object and allows you to specify the characteristics of an object.

The Business Process Modeling Notation (BPMN) complete set is supported by the Properties View. For example, if you are creating a gateway in your process diagram, you can select the type of gateway you want in the Properties View:



To see this view, select **Window > Show View > Properties** or **Window > Show View > Other**, then expand the **General** folder and select **Properties**. To view property information, you must have an object selected.

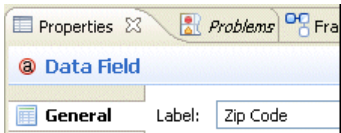


Properties related to simulation are only present when **Simulation** is selected as the destination environment.

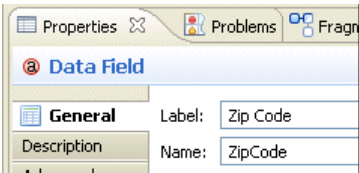
Properties related to the implementation of a process (for example, the **Input To Service** and **Output From Services** tabs for service tasks) are described in the appropriate implementation guide.

Labels and Names

When using the Business Analysis capability, process objects such as data fields, task names, and so on, have labels that may contain spaces or non-alphanumeric characters.



With the Solution Design capability selected, the Label as well as the Name is displayed. For example:



The Name is usually made up of the Label without whitespace or special characters. If you migrate a process from an earlier version, the migration XSLT creates Labels from the Name. The Name is used for referencing items where whitespace or special characters are restricted (for example, scripts and expressions). The Label is used for on-screen display.

If the Name has not been modified from the default, it automatically changes to match the Label when the Label is changed.

Process Properties

When you have selected a Process in the Project Explorer, the following properties are available.

Table 4: Process Property Reference

Tab	Property	Description
General	Label	Label associated with the process. You can change the label if necessary.
	Inline Sub-Process:	Select the Inline sub-process during Process Package Optimization checkbox so that at runtime, the objects such as activities and events contained in the reusable sub-process are brought into the top-level process and executed there, rather than by making a call to the sub-process. For more information, see Inline Sub-Processes (Package Optimization) on page 208 .
	Process Interface	Lists the location and name of the process interface implemented by the selected process. Specifying a process interface allows the dynamic selection of sub-processes at runtime. For more information, see Process Interfaces on page 218 .

Table 4: Process Property Reference (Cont'd)

Tab	Property	Description
	Used By	Lists any processes or tasks that use the selected process. For example, if you select a reusable sub-process the process and task that use the reusable sub-process are displayed. Click Go To to display the referencing object. This field helps prevent deleting a process that is used in another. Note, however that a process may be used by other packages, and these references cannot be displayed in this field.
Description		Optional textual description of the activity.
	Documentation Url	Allows you to specify a URL for supplementary information.
Destinations		Enter the target environment for the process. When you save the process, validation is performed according to this setting: • All processes are validated for BPMN 1.2 - additionally, other destinations can be selected to validate processes for use in specific environments • If you select BPM, the Process will be validated against BPM Version 1.0. • If you select Simulation, processes created under this project will be validated for simulation in TIBCO Business Studio as well as for BPMN correctness. • If you select iProcess, the Process will be validated against iProcess Engine Version 10.5 or higher. • If you do not select a destination environment, basic BPMN validation will be performed.
Resource	Separation Of Duties <i>(Manual and user tasks)</i>	Allows you to specify that certain manual or user tasks must be executed by different resources at runtime. For more information, see Separation of Duties on page 109 .
	Retain Familiar <i>(Manual and user tasks)</i>	Tasks within a Retain Familiar task group will, by preference, be offered to the same user.

Table 4: Process Property Reference (Cont'd)

Tab	Property	Description
Extended		Extended attributes can be used to add extra information to the schema. Click Add to add a new extended attribute. Modify the following: • Name - provide a Name for the attribute (you cannot include spaces in the name). • Value - add any text for the Value. • Escape Body - Select this checkbox if you want the parser to interpret the contents of the body attribute as text rather than XML (you do not want the parser to interpret symbols such as < or >). If you are entering XML, do not select this checkbox so the parser can validate the body attribute. • Body - enter whatever text or XML you want to make up the body of attribute.
Advanced		This tab may contain destination-specific properties.

Activity Properties

When you have selected an activity in the Process Editor, the following properties are available:

Table 5: Activity Property Reference

Tab	Property	Description
General	Label	Label for the activity that will be displayed on the diagram.
	Activity Markers	Select from Standard Loop , Multiple Instance Loop , or Ad-Hoc . For more information, see Activity Markers on page 102 .
	Participants	Specify the participant responsible for this activity (see Creating Participants on page 180).
	Activity Type	For example, Task , User Task , Service Task , and so on, (see Activity and Task Overview on page 92). If you select Reusable Sub-Process , you must browse to locate the sub-process.

Table 5: Activity Property Reference (Cont'd)

Tab	Property	Description
	- No Form URL - User Defined URL - Form - Pageflow	Configure the user task as follows: No Form URL Select this option if you do not want to display a form for the user task, or if upon export to iProcess you want to display a standard iProcess form. User Defined URL Select this option if you want to point to a specific URL (for example, if you authored a form outside of TIBCO Business Studio). Manually enter the URL. TIBCO Business Studio cannot validate the URL, so ensure it is correct. If you select this option but do not specify a URL, upon export to iProcess a standard iProcess form is created. This option is only applicable for iProcess. Form Use this option if you have created a form using TIBCO Business Studio Forms. Either automatically create a form (in which case the Form field is completed automatically) or browse to select a form from a Forms special folder. Upon deployment/import to the iProcess Engine, the task becomes a step with a Form type of Formflow Form. Pageflow Select this option if you want the user task to call a pageflow. Browse to select a pageflow process. When deployed, the specified pageflow process is run when the work item associated with the user task is opened. For more information about user task pageflow processes, see Working with User Tasks and Pageflow Processes on page 103.
	Reference Task <i>(Activities of type Reference Task)</i>	Shows the name of the task that is referenced from the currently selected task. Click Go To to highlight in the Process Editor the task to which the current task refers. Click Back to show the referring task.

Table 5: Activity Property Reference (Cont'd)

Tab	Property	Description
	Is a transaction (Activities of type Embedded Sub-Process)	Select the New Sub-Process is a transaction checkbox if you want the new sub-process to become a transaction and therefore be under transaction control (see Transactions on page 101).
	Chained Execution (Activities of type Independent or Embedded Sub-Process)	Specifies that the selected sub-process implements the chained execution resource allocation pattern (see Refactoring Sub-Processes on page 197).
	Sub-Process location (Activities of type reusable sub-process)	Sub-Process location is populated with the path to the Sub-Process name that you select.
	Sub-Process name (Activities of type reusable sub-process)	Browse to select the Sub-Process name .
	Runtime Identifier Field (Activities of type reusable sub-process)	Specify the field that will be used in the runtime process engine to dynamically select a sub-process that implements a process interface. For more information, see Process Interfaces on page 218 .
Description		Optional textual description of the activity.
	Documentation Url	Allows you to specify a URL for supplementary information.
Interface (all task types except Reference)	Visibility	Specify the visibility of an event or task (whether it is private or public). Public events or tasks publish information (such as required parameters) to an external process or application. Private events or tasks to publish any information to external applications.

Table 5: Activity Property Reference (Cont'd)

Tab	Property	Description
	Data	Allows you to select the data fields or parameters that the task requires as input and output. By default all process data is available to a task. When you explicitly associate process data with an Event or task, only that process data is associated with the event or task. After you add data fields, you can control whether it is an input parameter, output parameter, or both by setting its Mode (to In, Out, or In/Out). You can also specify whether the process data is mandatory. For more information, see Associating Process Data with Events and Tasks on page 173 Note that for user tasks, inbound and outbound parameters are from the perspective of the <i>form</i>, not the <i>user</i>. This means that inbound parameters are sent to the form by the process, not the user. outbound parameters are sent to the form by the user. For more information, see the appropriate implementation guide.
Resource	Participants <i>(Manual and user tasks only)</i>	This is the same as the Participant specified on the General tab.
	Initial Priority <i>(Manual and user tasks only)</i>	Specifies a relative priority for the work item. The affect and meaning of this setting is destination specific (refer to appropriate implementation guide).
	Distribution Strategy <i>(Message events and user tasks only)</i>	Allows you to specify a resource allocation pattern. For more information, see Resource Patterns and Work Distribution on page 108 .
	Piling <i>(Manual and user tasks only)</i>	
	Separation of Duties <i>(Manual and user tasks)</i>	Allows you to specify that certain manual or user tasks must be executed by different resources at runtime. For more information, see Separation of Duties on page 109 .

Table 5: Activity Property Reference (Cont'd)

Tab	Property	Description
	Retain Familiar (Manual and user tasks)	
Scripts		Allows you to add text that describes various types of scripts that you want added to a task. It is the task of the solution engineer to translate the outline of the script into a scripting language supported by the selected destination environment. For more information about implementing scripts, see the appropriate implementation guide.
Map To Sub-Process <i>(Activities of type reusable sub-process)</i>		Use this section to create a mapping from an actual parameter (data field or parameter) into the formal parameters of the reusable sub-process. Create a mapping by clicking a parameter, dragging to the destination parameter, then releasing the mouse button.
Map From Sub-Process <i>(Activities of type reusable sub-process)</i>		Use this section to create a mapping from a formal parameter of a reusable sub-process to an actual parameter (data field or parameter). Create a mapping clicking a parameter, dragging to the destination parameter, then releasing the mouse button.
Appearance	Line Color	Click the button next to Color to select the line color for the border of the selected activity. Click Set As Default For Type to apply your current color settings to any new activities of that type that you place on the process. You can revert to the default color settings by clicking Restore Factory Settings .
	Fill Color	Click the button next to Color to select the fill color for the selected activity. Click Set As Default For Type to apply your current color settings to any new activities of that type that you place on the process. You can revert to the default color settings by clicking Restore Factory Settings .
Extended		See the description of the Extended tab for the Process Properties on page 255 .

Table 5: Activity Property Reference (Cont'd)

Tab	Property	Description
Advanced	BPMN Task Extensions	Allows you to enter duration values for the task as per BPMN. How these durations are used is dependent on how the are implemented in a particular destination environment.

Gateway Properties

When you have selected a gateway in the Process Editor, the following properties are available:

Table 6: Gateway Property Reference

Tab	Property	Description
General	Label	Label for the gateway that will be displayed on the diagram.
	Gateway type	Choose the type of gateway (see Gateways on page 134 for more information).
	Discriminator Type <i>(Complex Gateway)</i>	Allows you to select a structured discriminator (see Complex on page 137).
	Parallel Split Name <i>(Complex Gateway)</i>	Allows you to specify the name of the upstream parallel gateway that this complex gateway is handling.
	Incoming Paths <i>(Complex Gateway)</i>	Specifies how many input Sequence flows are required before the output sequence flow is followed.
Description		Optional textual description of the gateway.

Table 6: Gateway Property Reference

Tab	Property	Description
Appearance		See the description of the Appearance tab for the Activity Properties on page 257 .
Extended		See the description of the Extended tab for the Process Properties on page 255 .
Advanced		Shows properties and values for the gateway.

Sequence Flow Properties

When you have selected a sequence flow in the Process Editor, the following properties are available:

Table 7: Sequence Flow Property Reference

Tab	Property	Description
General	Label	Label for the flow that will be displayed on the diagram.
	Type	Choose the type of Sequence Flow (see Sequence Flows on page 127 for more information).
Appearance		See the description of the Appearance tab for the Activity Properties on page 257 .
Description		Optional textual description of the Sequence Flow.
Extended		See the description of the Extended tab for the Process Properties on page 255 .
Advanced		Shows properties and values for the sequence flow

Event Properties

When you have selected an Event in the Process Editor, the following Properties are available:

Table 8: Event Property Reference

Tab	Property	Description
General	Label	Label for the event that will be displayed on the diagram.
	Trigger Type (<i>for Start or Intermediate events</i>) or Result Type (<i>for End events</i>)	You specify the trigger for a Start or Intermediate event and the result for an End event. The properties displayed on this page depend on the Trigger Type or Result Type you have selected. For more information about the types of events, see Working with Events on page 139 .
	Catch Error Code (<i>for catch error intermediate events only</i>)	Set to one of the following, depending on what error codes you want caught: • Catch All The event catches any error thrown by any event. • Catch By Name The event catches an error of the specified name, thrown by any activity. • Catch Specific The event catches a specific error thrown by a specific activity. For more information, see Using Throw and Catch Error Events on page 151.
Interface		See description for Activity Properties on page 257 .
Resource (<i>message event only</i>)	Distribution Strategy	See description for Activity Properties on page 257 .

Table 8: Event Property Reference (Cont'd)

Tab	Property	Description
Scripts		See description for Activity Properties on page 257 .
Map From Error <i>(for catch error intermediate events only)</i>		Use this section to create a mapping from error parameters to process data fields or parameters.
Appearance		See the description of the Appearance tab for the Activity Properties on page 257 .
Description		Optional textual description of the Event.
Extended		See the description of the Extended tab for the Process Properties on page 255 .
Advanced		

Lane Properties

When you have selected a Lane in the Process Editor, the following Properties are available:

Table 9: Lane Property Reference

Tab	Property	Description
General	Label	Label for the lane that will be displayed on the diagram.
Appearance		See the description of the Appearance tab for the Activity Properties on page 257 .
Extended		See the description of the Extended tab for the Process Properties on page 255 .
Advanced		This tab may contain destination-specific properties.

Cheat Sheets View

When you first start TIBCO Business Studio, a Welcome Cheat Sheet is displayed. You can view TIBCO Cheat Sheets and other Cheat Sheets by selecting **Help > Cheat Sheets** and selecting from the list.

Process Editor

The **Process Editor** is where you create your business process. It includes a [Palette](#) that contains the tools you use to create your Process.

Palette

You can create your Process using the tools on the palette. The Business Process Modeling Notation (BPMN) core set is supported by the palette, which contains the following tools:

Table 10: Palette Tool Reference

Palette Item	Description
 Select	Allows you to select objects.
 Marquee	Allows you to select several objects by drawing a box around them.
Favorites 	Displays some of the most common process objects. You can customize the Favorites drawer by right-clicking it and selecting Customize .
Connections 	Allows you to connect objects: • Sequence flow shows the order of objects in the process. • Conditional sequence flow is followed based on the evaluation of a condition. • Default sequence flow is followed if other conditions evaluate to false. • Message flow indicate the flow of messages between objects in separate Pools or between Pools. • Association either connects flow and non-flow objects or specifies the compensation task for a compensation event on a task boundary.

Table 10: Palette Tool Reference

Palette Item	Description
Start Events 	Indicates the beginning of the process as None, Message, Timer, Conditional, Multiple, or Signal. For more information, see Start Events on page 143.
Catch Intermediate Events 	Catches a "throw" event. The following Types are available: None, Message, Timer, Conditional, Link, Signal, Multiple, Error, Compensation, and Cancel. For more information, see Intermediate Events on page 144.
Throw Intermediate Events 	Throws an event. The following Types are available: Message, Compensation, Signal, Multiple and Link. For more information, see Intermediate Events on page 144.
End Events 	Indicates the end of the process. The following types are available: None, Message, Multiple, Error, Compensation, Cancel, Signal, and Terminate. For more information, see End Events on page 146.
Tasks 	Tasks of the following types: None, User, Manual, Service, Script, Send, Receive, Reference, Reusable Sub-Process, Embedded Sub-Process. For more information, see Activity and Task Overview on page 92.
Gateways 	Controls the flow of the process with the following types of gateways: Exclusive (Data), Parallel, Exclusive (Event), Inclusive, and Complex.

Table 10: Palette Tool Reference

Palette Item	Description
Artifacts 	Allows you to enhance the process with Text Annotations, Data Objects, Groups, Pools, and Lanes.

Alignment Tools

For easier alignment of objects in the Process Editor, you can select either or both of the following options:

- **Diagram > Grid** - This allows you to align objects to a grid.
- **Diagram > Alignment Guides** - As you move objects in a Process Editor, a blue line appears when the center of two objects coincide.

You can also align objects that you have selected by choosing options from the **Diagram > Alignment** menu. For example, if you have two Activities selected, you can align them horizontally on their centers by selecting **Diagram > Alignment > Align Middle**. Alternatively you can use the alignment buttons on the toolbar that correspond to the menu options.

Chapter 12 **Tips and Tricks**

This chapter contains tips for working with the TIBCO Business Studio user interface.

Topics

- [Process Editor, page 272](#)
- [Workbench, page 274](#)
- [Problems View, page 275](#)

Process Editor

The following tips relate to the Process Editor:

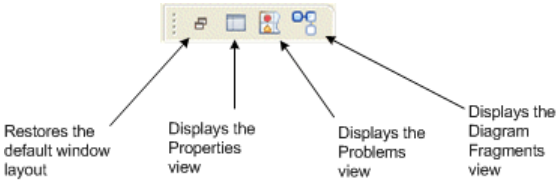
<i>I can't see the palette.</i>	Make sure you are in the Process Editor and that the palette is expanded (click the arrow to the right, top of the scroll bar).
<i>I have a large diagram that is difficult to view. How can I find a specific task?</i>	Press Ctrl+F and enter the name or a partial name in the resulting dialog. A list of search results is displayed. Clicking one of the results displays it in the Process Editor.
<i>How can I create several objects of the same type using the palette?</i>	Press the Ctrl key while adding the objects to your Process. For example, if you want to create several Sequence Flows, press and hold the Ctrl key while clicking the Flow tool and adding the Sequence Flows.
<i>How do I align objects in the Process Editor?</i>	There are two features that allow you to better align objects in the Process Editor - Grid and Alignment Guides. These are both available from the Diagram menu when using the Process Editor. Grid allows you to snap objects to the grid lines, and Alignment Guides shows a centering line when you have aligned two objects. With several objects selected in the Process Editor you can also select options from the Diagram > Alignment menu to control the vertical and horizontal alignment of the objects.
<i>I'm using Grid alignment in the Process Editor and I want to place an object without using the Grid.</i>	Pressing the Alt key while moving an object allows you to position it between grid lines.

<i>I want to see a larger view of my Process.</i>	You can expand the Process Editor to fill your screen by double-clicking its title bar or pressing Ctrl+M. You can temporarily display the Properties view by double-clicking any diagram element. The Properties view will be hidden again when you click on the diagram.
<i>How can I easily resize an Activity in the Process Editor?</i>	Pressing the Ctrl key while resizing one side of an object automatically resizes the opposite side as well.
<i>I can't get my Sequence Flows to look right.</i>	Occasionally when you reposition a Sequence Flow several times you create too many bendpoints and make the flow illegible. To restore a Sequence Flow, click the flow and select Reset BEndpoints .
<i>How can I create bendpoints when I draw a Sequence Flow?</i>	Whilst holding down the left mouse button and positioning the Sequence Flow, right-click to add a bendpoint.
<i>I can't get a Sequence Flow to dock where I want it to dock.</i>	Sequence Flows are docked at the nearest docking point rather than where you attempt to dock them. Try repositioning the middle of the Sequence Flow to change the docking point.
<i>How can I position Sequence Flows using the keyboard?</i>	Highlight the Sequence Flow, then press the Period key (.) to move along the bendpoints. With a bendpoint highlighted, use the arrow keys to position them and change the shape of the flow. When you are done, press the Return key.
<i>How can I email someone a Process I am working on?</i>	Send them the XPD L Package file that contains the Process. Locate the Package file name in the Project Explorer. The Package file is stored in your workspace (by default this is C:\Documents and Settings\username\workspace).

Workbench

The following tips relate to the Eclipse Workbench:

<i>How can I quickly change Perspectives?</i>	There are buttons in the upper right of the screen so you can change between the Simulation and Modeling perspectives. You can also do this using Ctrl+F8.
<i>My windows are a mess. How can I reset them?</i>	Select Window > Reset Perspective . This restores the current Perspective to its default.
<i>How can I switch to a different workspace?</i>	Select File > Switch Workspace and browse for the location of the workspace. Note that after you select a workspace and click OK , TIBCO Business Studio must restart before you can access the new workspace.
<i>After starting TIBCO Business Studio, I get the following error message:</i> Unable to read workbench state. workbench UI layout will be reset	This error occurs because a UI element (for example, editor, view, or perspective) that was displayed at the close of the last TIBCO Business Studio session is no longer available. This error usually indicates a change in the installation environment (a product has been removed or upgraded), and can safely be ignored.
<i>How can I navigate to other views while in full screen mode?</i>	When a view or editor is expanded to fill the window, Eclipse provides several buttons around the border of the screen that allow you to display other views in addition to the expanded view of the Process. For example, the following buttons are displayed around the border of the screen:



Problems View

The following tips relate to the Problems view:

How can I correct validation errors?

Either:

- Right-click the problem and select **Quick Fix** (if enabled for the current problem). This gives you the option of having TIBCO Business Studio automatically correct the problem for you.
- or -
- If no quick fix is available, Double-click the problem or right-click the problem and select **Go To**. This displays the Process in the Process Editor, highlighting the offending object and allowing you to manually correct the problem.

In the Problems view I see not only problems for my Process/Project, but problems for all Processes/Projects.

From the menu in the upper right of the Problems view, select **Configure Contents**. This allows you to customize the problems display. For example, you can select options to display problems for the selected resource only, or for any resource in the same Project.

Index

A

Activity [92](#)
 marker [102](#)
 Participant [182](#)
 properties [257](#)
 script [98](#)
 ARIS software
 creating custom symbols [240](#)
 custom symbols [240](#)
 export [241](#), [241](#)
 import [242](#)
 integrating with [240](#)
 object mapping [244](#)
 Asset [56](#)
 ad-hoc [57](#)
 Business Objects Model [58](#)
 types [11](#)
 Asset Central [74](#)
 Association [118](#)

B

Business Objects Model [58](#)
 Business Process Modeling Notation (BPMN) [12](#)

C

capability [13](#)
 Catch Error Event [149](#)
 Chained Execution [197](#)
 changes from the previous release TIBCO Business
 Studio Modeling User's Guide [xiv](#)
 Cheat Sheets View [266](#)
 customer support [xix](#)

D

Data Field [164](#)
 Data Object [118](#)
 Declared Type [172](#)
 Destination environment [236](#)
 Diagram Fragments View [253](#)

E

Eclipse
 capability [13](#)
 perspective [13](#)
 workbench [274](#)
 Emailing Processes [80](#)
 Embedded Sub-Process [194](#), [197](#)
 creating [197](#)
 End Event [146](#)
 Error Event [149](#)
 errors [236](#)
 Event [140](#)
 properties [264](#)
 result [140](#)
 trigger [140](#)
 visibility [178](#)
 exporting from ARIS [241](#)
 Extended Attributes [51](#)

F

Formal Parameter [164](#)
 Fragment [70](#), [70](#)

G

Gateway [134](#)
 properties [262](#)
 graft steps [232](#)
 Group [120](#)

I

importing ARIS procedures [242](#)
 Inline Sub-Process [209](#)
 criteria [215](#)
 Intermediate Event [144](#)

L

Lane [121](#)
 properties [265](#)
 loop
 multi-instance [129](#)
 standard [128](#)

M

Manual Task [93](#)
 Message Event [158](#)
 Message Flow [126](#)
 Migration [37, 41](#)

O

Outline View [252](#)

P

Package [9, 9](#)
 copying [41](#)
 creating [27](#)
 Palette [267](#)
 using [27](#)
 Parameter [164](#)
 Participant
 associating with Activity [182](#)
 creating [180](#)
 perspective [13](#)
 Pool [123](#)
 private event [178](#)
 private task [178](#)
 Problems View [251, 275, 275](#)
 Process [9, 9](#)
 adding objects to [27](#)
 correcting errors [236](#)
 deleting [42, 49](#)
 properties [255](#)
 re-use [69, 118, 149, 190](#)
 selecting objects [32](#)
 sending in email [80](#)
 working with [27](#)
 Process components [5](#)
 Process documentation [83](#)
 Process Editor [267, 272](#)
 alignment tools [269](#)
 tips [272](#)
 using [27](#)
 Process Interface
 creating [221](#)
 modifying [231](#)
 Process Package documentation [81](#)
 viewing [83](#)
 Process Properties [255](#)
 Project [9, 9](#)
 creating [16, 22](#)
 creating from source control [73](#)
 purpose [16, 22](#)
 Project Explorer View [250](#)
 Properties View [254](#)
 public event [178](#)
 public task [178](#)

Q

quality process [56](#)
 Quick fix [236](#)

R

Receive Task [94](#)
 refactoring Sub-Processes [197](#)
 Reference Task [95](#)
 resource allocation [108](#)
 resource pattern [197](#)
 Reusable Sub-Process [199](#)

S

Script Task [93](#)
 Send Task [94](#)
 Separation of duties [109](#)
 Sequence Flow [127](#)
 properties [263](#)
 using to connect objects [30](#)
 Service Task [93](#)
 source control [72](#), [73](#)
 Special Folder
 enabling and disabling [59](#)
 Standards Support [12](#)
 Start Event [143](#), [221](#)
 Sub-Process [100](#)
 call to a Process Interface [228](#)
 expanding [193](#), [193](#)
 inline [209](#)
 refactoring [197](#)
 Subversion [72](#)
 support, contacting [xix](#)
 SVN repository
 browsing [73](#)
 deleting Project from [74](#)
 Swimlanes [121](#)

T

Task [92](#)
 properties [257](#)
 script [98](#)
 visibility [178](#)
 technical support [xix](#)
 Text Annotation [119](#)
 Throw Error Events [149](#)
 TIBCO Business Studio
 using with Subversion [72](#)
 TIBCO_HOME [xvii](#)
 Timer Event Script [147](#)
 transactions [101](#)

U

User Task [92](#)

V

Validation [236](#)
 validation errors [236](#)

X

XML Process Definition Language (XPDL) [12](#)